Clusters in Aggregated Health Data *

Kevin Buchin, Maike Buchin, Marc van Kreveld, Maarten Löffler, Jun Luo, and Rodrigo I. Silveira

Department of Information and Computing Sciences, Utrecht University, the Netherlands; email: {buchin, maike, marc, loffler, ljroger, rodrigo}@cs.uu.nl

Abstract. Spatial information plays an important role in the identification of sources of outbreaks for many different health-related conditions. In the public health domain, as in many other domains, the available data is often aggregated into geographical regions, such as zip codes or municipalities.

In this paper we study the problem of finding clusters in spatially aggregated data. Given a subdivision of the plane into regions with two values per region, a case count and a population count, we look for a cluster with maximum density. We model the problem as finding a placement of a given shape R such that the ratio of cases contained in R to people living in R is maximized. We propose two models that differ on how to determine the cases in R, together with several variants and extensions, and give algorithms that solve the problems efficiently.

Keywords: cluster, outbreak, algorithm, aggregated data.

1 Introduction

The study of geographical patterns of diseases is an important aid for the investigation of outbreaks. Analyzing the geographic nature of disease cases has been a key factor in finding the source of many outbreaks. The classical example is the outbreak of cholera in the Soho district of London in 1854 [2, 21]. The source of this outbreak, that left a death toll of more than 600 people in about 10 days, was found by John Snow, a London physician. He realized that most affected people lived around a public pump, which was later confirmed as the source. Numerous other examples have been documented since then in the literature of several fields like epidemiology, public health, preventive medicine, and medical geography.

* This research was initiated during the GADGET Workshop on Geometric Algorithms and Spatial Data Mining, funded by the Netherlands Organisation for Scientific Research (NWO) under BRICKS/FOCUS grant number 642.065.503, and has been also partially funded by NWO under the project GOGO.

Investigation of outbreaks due to both infectious and noninfectious causes (e.g., toxic exposure) can greatly benefit from the use of spatial information. Even though the role played by geography in the identification of the source depends entirely on the disease, spatial factors have a major importance for many point source outbreaks related to exposure to pollution or radiation sources (for a wide range of diseases, from respiratory illnesses to different types of cancer), as well as for airborne diseases like Legionella [6] or Q fever [8]. As these examples show, for many types of diseases, finding the source of the outbreak can be seen as a spatial data clustering problem.

In general, clustering is the process of grouping objects into meaningful subclasses (that is, clusters) so that the objects within a cluster have high similarity in comparison to one another, but are dissimilar to objects in other clusters [9, 10, 11]. Spatial clustering deals with spatial objects, and disease clustering in particular deals with (geographically referenced) cases of a disease or another health-related condition. The problem of identifying the source of an outbreak can be seen as finding a location whose neighborhood has a disease rate that is *significantly* higher than for other locations. However, the problem studied here differs from the standard spatial clustering problem in two ways.

Firstly, typical clustering algorithms consider only the absolute number of objects within clusters (that is, the density of the cluster is defined as the case count), or, in the case of density-based clustering algorithms, the number of points per unit area [7, 1]. However, in disease clustering the number of cases is not very meaningful if it does not consider the population-at-risk. For example, considering only the absolute number of cases does not take into account that the population can be clustered itself within an urban area. To adjust the case count for the population density, we define the density of the cluster as the ratio of cases to exposed people, that is, we look for clusters with high density (or *attack rate*, in epidemiology).

Secondly, in the traditional clustering problem the exact location of the cases is known. This can be the case in disease clustering, as in the two examples mentioned before. Still, in many situations the precise case location is not available. On the one hand, it is common to have a data source that does not include this information. Statistics data is very often aggregated into areas corresponding to regions like counties, zip codes, census blocks or enumeration districts, or come from sources like anonymous questionnaires where only approximate locations (like partial zip codes) are provided in the first place. On the other hand, even if the data is available, there are privacy and confidentiality considerations for not disclosing exact address information of patients [5, 4]. Although this paper focuses on aggregated data in public health, it is worth mentioning that aggregated data is also frequently used in other areas such as criminology [17], sociology [19], political science [3, 12], and geography [15].

In the public health domain, aggregated data clustering is done by statistical methods. One of the most widely used approaches for cluster detection

for disease surveillance is the *spatial scan statistic* of Kulldorff and Nagarwalla [14, 13].

But they represent the aggregation regions by points; thus, the spatial scan statistic does not directly handle aggregated data. Furthermore, the candidate cluster regions (windows) are positioned only at grid points (of a predetermined grid), which simplifies the problem. Another well-known method for cluster detection is by the Geographical Analysis Machine [16]. This method assumes non-aggregated point data and only tests cluster regions based on grid points as well.

This paper begins by modeling the problem as a rectangle placement problem (a rectangle is chosen for illustration; the ideas apply equally well to a square or regular 10-gon, for example, which allows us to approximate a circular cluster region). We first present a simple model that assumes uniform distribution of both the cases and the population, and then we provide a second model with a different density measure. Some possible extensions of these models are also discussed. We then present an algorithm for the first model. It computes a location for a cluster center by considering *all* possible placements of a rectangle R over a subdivision with n regions. This is an important difference to previous approaches, which restrict the search to a finite set of points. Our algorithm is based on computing the arrangement of the combinatorially different placements of R, and on optimizing a density function within each cell of the arrangement. The total worst-case running time of the algorithm is $O(n^2)$, but we prove that under reasonable, practical assumptions on the resolution of the regions and R, the running time is only $O(n \log n)$. The algorithm is flexible enough to allow extensions to several variations. After explaining the algorithm for the first model in detail, we discuss how to adapt it for the second model, and how to incorporate variants like different shapes for the cluster region, or having two different subdivisions for the case and population data. All these variations can be easily inserted into the algorithm for the first model, although sometimes at the expense of an increase in the running time.

2 Model

In this paper we abstract the problem of finding the source of a point source outbreak as a rectangle placement problem. In the models proposed next, we are given a subdivision of the plane, consisting of a set $\mathcal{P}$ of n regions $P_1, \ldots, P_n$, and for each region P_i in the subdivision we are given two values c_i and p_i. The first value c_i represents the number of disease cases within P_i, whereas the second value p_i represents the population of P_i (for example, the number of people at risk for the disease in question).

The outbreak area is modeled by a rectangle R, and the objective is to find a placement of R such that the *density* of cases covered by the rectangle is maximized. The density in R will be defined in the following models. We

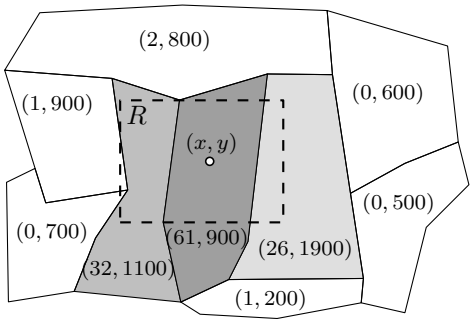

Fig. 1. Example for model I: for each region P_i, (c_i, p_i) is shown; shading visualizes density. The cases and population are assumed to be uniformly distributed inside each region. The goal is to place a rectangle R such that the density of R is maximized.

assume that we have access only to aggregated location data, meaning that the exact location of the cases is not known. The cluster rectangle R will have some fixed size and we will assume that it is axis-aligned.

We propose two basic models. The first model assumes that the distribution of the cases is *uniform* inside each region. The second model assumes a *worst-case* distribution of the cases, that is, all cases of regions intersected by R are assumed to appear inside the rectangle. Moreover, some possible variants, with different subdivisions for the case and population data, and different shapes for the outbreak area, are discussed at the end of this section.

Model I We will assume for the first model that the distributions of both the cases and the population are uniform. The density of the rectangle R is defined simply as the number of cases inside R divided by the total population in R.

More formally, let (x, y) be the position of the center of rectangle R. We will write $R(x, y)$ to refer to the rectangle R translated in such a way that its center lies at (x, y). See Figure 1 for an illustration. Finding a placement of R is equivalent to finding a value for x and y. The goal can be expressed as:

$$\max_{(x,y)\in\mathbb{R}^2} \frac{\sum_{i=1\ldots n} c_i \cdot f_i(x, y)}{\sum_{i=1\ldots n} p_i \cdot f_i(x, y)} \tag{1}$$

where $f_i(x, y) \in [0, 1]$ denotes the fraction of the area of P_i intersected by $R(x, y)$: $f_i(x, y) = Area(P_i \cap R(x, y))/Area(P_i)$.

This model seems reasonable given that we are dealing with aggregated data and the location of the cases is not known. However, there are situations in which the result obtained is not what one would expect. As an example, consider the example depicted in Figure 2. Under model I, the optimal rectangle

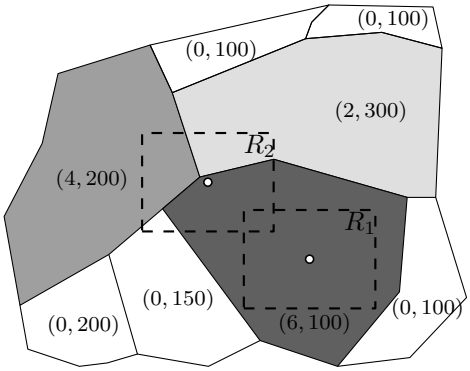

Fig. 2. The numbers in each region indicate number of cases and population. Model I will result in a rectangle like R_1, whereas model II will yield one like R_2.

lies completely inside the most dense region, like R_1. However, the situation suggests that the source of the outbreak must be close to the intersection of the three shaded regions. The next model we propose handles such situations better.

Model II As illustrated by Figure 2, the previous model does not always give the most reasonable result. In a situation like the one shown, if the real outbreak source is close to the meeting point of the three shaded regions, the assumption of uniform distribution of the cases within the regions is not valid.

To try to deal with this situation, we propose a second model where we assume that for a given location of R, all cases in each region intersected by R are concentrated inside R. This can be seen as a *worst-case* density measure for R, in accordance with the idea that most of the cases in a point source outbreak will be concentrated around the source. If the fraction of a region intersected by R is too small, we run the risk of counting more cases in a region than the number of people living in the intersection with R. To avoid this, we will take the minimum between the case count and the number of people assumed to live within that fraction of the region.

We formalize this model in the following way:

$$\max_{(x,y)\in\mathbb{R}^2} \frac{\sum_{i=1\ldots n} \min\{c_i,\ p_i \cdot f_i(x,y)\}}{\sum_{i=1\ldots n} p_i \cdot f_i(x,y)} \tag{2}$$

Two-subdivision variant It may be that the population information and the case information are aggregated differently. Then we would have one subdivision for the population data and another one for the number of cases. See Figure 3 for an example.

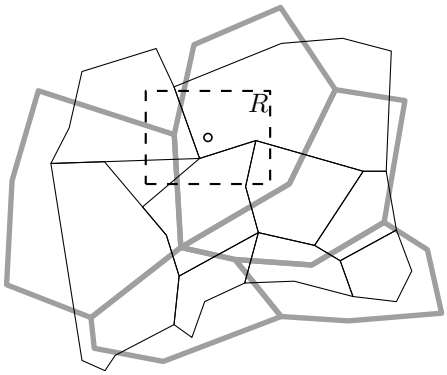

Fig. 3. A variant of the problem is where two different subdivisions are given, one for the case data and one for the population data.

We are now given a subdivision $\mathcal{P}$ of the plane comprised of n regions $P_1, \ldots, P_n$, and for each region P_i we are given a population value p_i, and in addition, we are given a second subdivision $\mathcal{C}$ comprised of m regions $C_1, \ldots, C_m$, each with a case count value c_i.

Both models I and II can be used for this variant. The equation for model I could be expressed as:

$$\max_{(x,y)\in\mathbb{R}^2} \frac{\sum_{j=1\ldots m} c_j \cdot g_j(x,y)}{\sum_{i=1\ldots n} p_i \cdot f_i(x,y)} \tag{3}$$

where $g_j(x,y) \in [0,1]$ denotes the fraction of the area of C_j intersected by $R(x,y)$, and $f_i(x,y) \in [0,1]$ denotes the fraction of the area of P_i intersected by $R(x,y)$. Note that in some unlikely cases, this measure can yield a value greater than 1, so it is not a real 'density' measure. Adaptations that address this issue are possible: the algorithm described in the next section is flexible enough to allow for a wide range of measures. In practice, however, the number of cases is expected to be much lower than the population count, and this should not be a problem.

For model II, the goal could be reformulated as:

$$\max_{(x,y)\in\mathbb{R}^2} \frac{\sum_{j=1\ldots m} \min\{c_j, \sum_{i=1\ldots n} p_i \cdot g_{ij}(x,y)\}}{\sum_{i=1\ldots n} p_i \cdot f_i(x,y)} \tag{4}$$

where $g_{ij}(x,y) \in [0,1]$ denotes the fraction of the area of P_i intersected by $R(x,y)$ and C_j, and $f_i(x,y) \in [0,1]$ denotes the fraction of the area of P_i intersected by $R(x,y)$.

Different shapes for R In the previous model the outbreak area R was modeled with a rectangle (mainly because our algorithms are easier to describe in this case). However, depending on the characteristics of the disease under consideration, other shapes can be more appropriate. For example, for studying cases of exposure to some radiation source where the Euclidean metric applies, a disc can be better suited. For airborne diseases, when wind information is available, an ellipse with a certain rotation of the main axis can be a better choice. Both discs and ellipses (if approximated by a polygon) are variants of the outbreak area to which our algorithms can be extended, as discussed in the next section.

Resolution assumption In the analysis of the algorithm in the next section, we will not only consider the general worst-case scenario, but also a more realistic scenario. In particular, we will make a *resolution assumption*. Define the *resolution unit* r as the shortest distance between any two vertices of the region subdivision $\mathcal{P}$. Our *resolution assumption* states that there are positive constants c_1, c_2, c_3, c_4 such that (i) the distance between any vertex and any line segment not incident to that vertex is at least $c_1 r$, (ii) the length of any line segment in the subdivision is at most $c_2 r$, and (iii) the diameter of R is at least $c_3 r$ and at most $c_4 r$.

The assumption essentially states that the difference in scale between the regions, and between the rectangle R and the region subdivision are reasonable. For example, it would be very impractical to have regions that are city neighborhoods with an outbreak region of the size of the whole country. This assumption will allow to prove that in practice, the algorithms have a considerably better running time than what is provable otherwise.

Lemma 1. *The resolution assumption implies that any angle between two segments of $\mathcal{P}$ is bounded from below by a positive constant.*

Proof. Let v be a vertex of $\mathcal{P}$, and suppose there are two line segments with angle α that have v as an endpoint, and let the shorter have length l. Then the distance d between the endpoint of the shorter of the two and the longer segment will be $d = l \sin \alpha$. But we know that $l \leq c_2 r$ and $d \geq c_1 r$, which implies that $\sin \alpha \geq \frac{c_1}{c_2}$. Since c_1 and c_2 are positive constants, the lemma follows.

3 Algorithms

To solve the problems defined in the previous section, we will compute the arrangement of combinatorially different placements of the query rectangle R. The next subsection details what this arrangement looks like, and how to compute it efficiently. In the subsection that follows, we will use the arrangement to compute the optimal placement of the rectangle. Finally, we will describe how the method can be adapted to work for the other models.

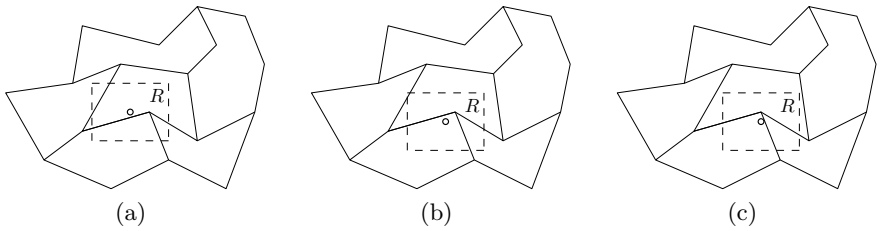

Fig. 4. Placements in (a) and (b) are are combinatorially the same, but in (c) it is combinatorially different: the set of regions intersected by R is different.

3.1 Arrangement of placements

Given a subdivision $\mathcal{P}$ and a rectangle R, we say two placements of R are *combinatorially different* if the set of pairs of edges of R and $\mathcal{P}$ that intersect are different. For example, the placements in Figures 4(a) and 4(b) are combinatorially the same, but the one in Figure 4(c) is different because the top left corner of R has moved from one cell to another. When two placements are the same, we can write the area of overlap between R and any cell $P \in \mathcal{P}$ as a closed-form function in x and y, which will allow us to optimize functions that involve this area of overlap efficiently for all combinatorially equal placements.

This combinatorial relation between placements subdivides the *placement space*—the set of possible positions for the reference point of R—into a number of regions such that inside each region, all placements are combinatorially equal. We can define and compute this arrangement for each cell $P \in \mathcal{P}$, and the total arrangement will just be the overlay of these. For a given cell P, a placement of R changes whenever a corner of R moves across an edge of P (as in Figure 4), or when a corner of P moves across an edge of R. This means the boundaries of the arrangement are exactly the line segments that arise when sliding a corner of R along an edge of P, as in Figure 5(b), or vice versa, as in Figure 5(c). The arrangement of this region is then the overlay of those two, as shown in Figure 5(d).

If we compute this arrangement for all cells of the subdivision, and compute the overlay of all of them, then this gives a new subdivision of the whole plane, such that within any cell the combinatorial structure does not change. To compute the arrangement, we collect all translated copies of the cells and of R, and note that their total number of vertices is $O(n)$ (as long as R has constant complexity). We can compute the overlay of all these polygons in $O(n \log n + k)$ time using standard methods, where k is the complexity of the final arrangement. In the worst case, this complexity can be $O(n^2)$. However, under the resolution assumption in Section 2, we can prove that the complexity is actually $O(n)$.

Lemma 2. *The complexity of the arrangement under the resolution assumption is $O(n)$.*

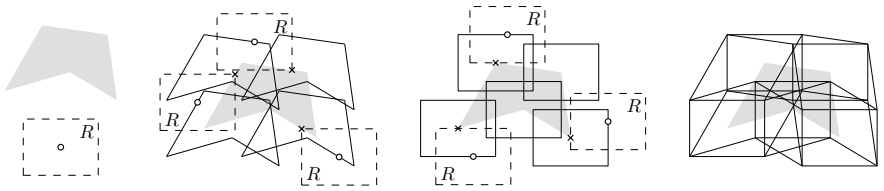

Fig. 5. (a) A cell of the subdivision (shaded), and a query rectangle R (dashed). (b) The positions of the reference point of R as a vertex of the rectangle slides along an edge of the cell. (c) The positions of the reference point of R as an edge of the rectangle slides along a vertex of the cell. (d) The total arrangement is the overlay of the previous two figures.

Proof. First, we show that R can never intersect more than a constant number of line segments of $\mathcal{P}$. Let V be the set of vertices inside R. We know that any two vertices are separated by at least $c_1 r$, and that the diameter of R is at most $c_4 r$. This means that the size of V can be at most $O((\frac{c_4}{c_1})^2)$ by a packing argument. Consider the set of line segments that intersect R, but do not have an endpoint in V. These segments must have a distance of at least $c_1 r$, and completely go through R, so there can be at most $O(\frac{c_4}{c_1})$ of them. By Lemma 1 a vertex $v \in V$ can be the endpoint of at most a constant number of line segments, so the total number of segments intersected by R is also constant.

Let p be any point in the plane. The rectangle R centered at p intersects at most a constant number s of features. This means p can be inside at most $O(s)$ different curves of the arrangement. Then we note that the regions in the arrangement corresponding to disjoint segments are pseudodiscs: the boundaries of two such regions cannot intersect more than twice. It is known that an arrangement of pseudodiscs with constant bounded depth has linear complexity [20].

3.2 Computing the optimal placement

To optimize Formula (1) over the arrangement, we first need to be able to optimize it inside a single cell. The only part of the formula that depends on x and y is the fraction $f_i(x, y)$ that describes which part of each region P in $\mathcal{P}$ is covered by R. The area of overlap can be decomposed into trapezoids, see Figure 6. The locations of the corners of these trapezoids are linear functions in x and y, so the area of each trapezoid is a quadratic function. We can then add up these functions, so that Formula (1) is in this form:

$$\max_{(x,y)\in\mathbb{R}^2} \frac{a_1x^2 + a_2xy + a_3y^2 + a_4x + a_5y + a_6}{b_1x^2 + b_2xy + b_3y^2 + b_4x + b_5y + b_6} \tag{5}$$

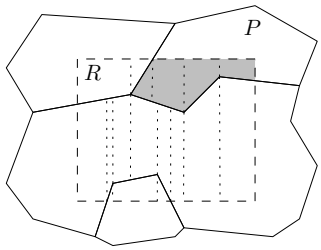

Fig. 6. The area of overlap between R and a subdivision cell P is the union of four trapezoids.

This formula can be optimized in constant time by using standard algebraic methods, or it can be numerically approximated very fast.

To find the maximum over the whole arrangement, we need to determine Formula (5) for every cell. We can of course just do this from scratch for each cell individually, but without the resolution assumption, that would require $O(n)$ time per cell, leading to a total of $O(nk)$ time (where k is the complexity of the arrangement). Instead, we will traverse the cells of the arrangement from neighbor to neighbor while maintaining some information. We maintain the numerator and the denominator of Formula (5) separately, and update them both when we move the reference point over the arrangement to a neighboring cell. Recall that the topological structure changes when a corner of R moves over an edge of P, or vice versa. Using some ideas from Reinbacher *et al.* [18], we can update the numerator and denominator in constant time when we move to a neighboring cell, basically by subtracting the contribution of quadratic functions that no longer give a trapezoid, and adding the contribution of quadratic functions that give a new trapezoid. Therefore, we spend only $O(k)$ time to determine Formula (5) for all cells, and to find the maximum. With the resolution assumption, this implies that we spend only $O(n)$ time in total.

3.3 Extensions

To solve the problem in the *second model,* our arrangements become a bit more complicated, because there is another event where the functions involved change: when the query rectangle starts containing enough area to allow all disease cases of some region to be inside it. This happens when the area of overlap between R and some region P_i becomes more than some fixed value: $Area(P_i) \cdot c_i/p_i$. This means we must add some extra curves to the placement space: the curves where the area of overlap has exactly this value. Generally this gives one closed curve (as R moves around P_i, keeping the area of overlap constant), but it could also be a collection of curves. Figure 7(a) shows how this looks in our example. In fact, the points where the pieces of this curve change coincide with the lines of the other parts of the arrangement, since this happens exactly when the combinatorial structure of the area of

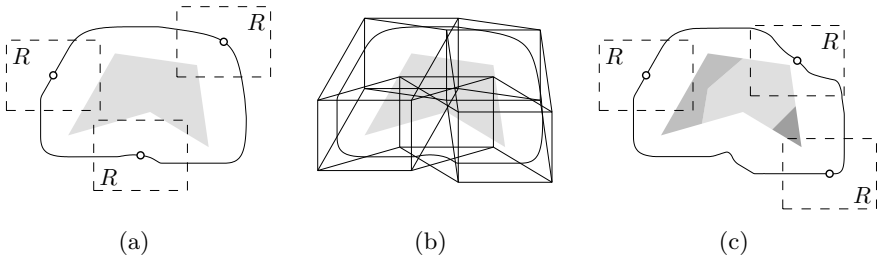

Fig. 7. (a) New curves introduced in the second model. (b) Total arrangement. (c) Non-uniform population density.

overlap changes. Within one piece, the curve is a level-set (or iso-contour) of a quadratic function, so it is a quadratic curve in the plane. Figure 7(b) shows the total arrangement. The functions we need to optimize over the new arrangement remain the same, only now we have to optimize them over cells with nonlinear boundaries. Standard numerical methods can be used to solve this problem.

When the information about the population and the disease cases are given in *separate subdivisions* $\mathcal{P}$ *and* $\mathcal{C}$ (in the first model), we can compute the overlay of the two and treat this as if it was a single subdivision. The algorithm still works without changes, but in the worst case the running time becomes as bad as $O(n^4)$. However, this will hardly occur in practice. In fact, under the resolution assumption in Section 2 we can prove that the complexity also stays linear.

Lemma 3. *In the two-subdivision variant of the problem, under the resolution assumption, the complexity of the arrangement is* $O(n)$.

Proof. Let l be a segment of $\mathcal{P}$. We will show that l intersects at most a constant number of segments of $\mathcal{C}$. This then implies that the overlay has linear complexity, and we can simply apply Lemma 2.

We know that the length of l is at most $c_2 r$. Sort the segments of $\mathcal{C}$ that intersect l. If two consecutive segments do not share an endpoint, then the distance between them is at least $c_1 r$. If they do share an endpoint, then this point must be at least $c_1 r$ away from l, and the angle between the segments, by Lemma 1, is at least $\arcsin \frac{c_1}{c_2}$, so the distance between them is at least some constant times r. Therefore the total number is constant.

When we have *separate subdivisions* for the population and the disease cases in the *second model,* we can still compute the overlay of the subdivisions, but now we need to be aware of the total number of disease cases in a certain *collection* of cells. Since we are assuming all cases inside a cell C are in the worst possible position, we cannot just distribute them evenly over the

smaller cells in the overlaid arrangement. Instead, we compute the curves of constant overlap directly for C, while taking the finer population subdivision into account. Figure 7(c) shows an example of this situation. The curve is still piecewise quadratic, only the number of pieces now also depends on the number of smaller cells.

When our query region R is not a rectangle, but some other constant size polygon, the algorithm still works without any modifications. The number of vertices of R will appear in the running time, but not asymptotically if it remains constant.

4 Discussion

In this paper we apply computational geometry tools to solve certain disease cluster problems on aggregated data. We presented models and algorithms for finding the densest cluster in spatially aggregated data. It can be seen as an aid for finding a likely source of disease outbreaks. One model comes down to placing a rectangle such that the ratio between the cases contained within the rectangle and the population in it is maximized. The proposed algorithm solves the problem in $O(n^2)$ time, and under realistic input assumptions on the resolution of the input, in $O(n \log n)$ time. A second model uses a different assumption on the distribution of the cases within a region, and several variants (like different shapes for the cluster region or case/population data in different subdivisions) are also discussed, showing how the algorithm can be extended for those cases.

The problem addressed differs from the more traditional cluster location problems in that we do not work with the exact positions of the points but with aggregated data. As explained before, aggregated data is very often used in public health and other domains. Most of the spatial clustering algorithms do not take aggregation of the data into account. Our approach also differs from the more traditional approaches used for spatial disease clustering because we do not restrict the search of possible locations for the rectangle to a finite subset of points (like the centroids of the regions), but effectively consider all possible placements.

Several problems remain open and constitute interesting topics for further research. One of them is to incorporate more advanced density measures, to be able to use, for example, a likelihood ratio test like the one used by Kulldorff and Nagarwalla [14] for a suitable statistical model. The main difficulty lies in being able optimize such a function over a cell of the arrangement of different placements. Another interesting extension to consider is when the case data comes from different sources, for example emergency department visits and over-the-counter medication sales. Then the challenge would be not only to combine the different geographic subdivisions efficiently, but also to account for possible double-counting of the cases. Thirdly, cluster detection

that includes the temporal component gives rise to new models of density where the time development of the cases in all regions may be known.

References

1. R. Agrawal, J. Gehrke, D. Gunopulus, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *Proc. ACM-SIGMOD Intl. Conf. on Mgmt. of Data*, pages 94–105, 1998.
2. H. Brody, M. R. Rip, P. Vinten-Johansen, N. Paneth, and S. Rachman. Map-making and myth-making in Broad Street: the London cholera epidemic, 1854. *The Lancet*, 356:64–68, 2000.
3. N. Cleave, P. Brown, and C. Payne. Methods for ecological inference: an evaluation. *Journal of the Royal Statistical Society, Series A*, 158:55–75, 1995.
4. L. H. Cox. Protecting confidentiality in small population health and environmental statistics. *Stat. Med.*, 15:1895–1905, 1996.
5. E. Cromley and S. McLafferty. *GIS and Public Health.* The Guilford Press, New York, 2002.
6. J. W. Den Boer, L. Verhoef, M. A. Bencini, J. P. Bruin, R. Jansen, and E. P. Yzerman. Outbreak detection and secondary prevention of legionnaires disease: A national approach. *International Journal of Hygiene and Environmental Health*, 210:1–7, 2007.
7. M. Ester, H. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. 2nd International Conference on Knowledge Discovery and Data Mining*, pages 226–231, 1996.
8. A. Gilsdorf, C. Kroh, S. Grimm, E. Jensen, C. Wagner-Wiening, and K. Alpers. Large Q fever outbreak due to sheep farming near residential areas. *Accepted for publication to Epidemiol. Infect.*, 2007.
9. J. Han and M. Kamber. *Data Mining: Concepts and Techniques.* Academic Press, San Diego, 2001.
10. J. Hartigan. *Clustering Algorithms.* John Wiley & Sons, New York, 1975.
11. A. Jain and R. Dubes. *Algorithms for Clustering Data.* Prentice Hall, Englewood Cliffs, New Jersey, 1988.
12. G. King. *A Solution to the Ecological Inference Problem.* Princeton University Press, Princeton, New Jersey, 1997.
13. M. Kulldorff. A spatial scan statistic. *Communications in Statistics: Theory and Methods*, 26,:1481–1496, 1997.
14. M. Kulldorff and N. Nagarwalla. Spatial disease clusters: detection and inference. *Stat. Med.*, 14:799–810, 1995.
15. S. Openshaw. *The Modifiable Areal Problem.* CATMOG No.38. Geo Books, Norwich, 1984.
16. S. Openshaw, M. Charlton, C. Wymer, and A. Craft. A Mark 1 Geographical Analysis Machine for the automated analysis of point data sets. *Int. J. Geographical Information Systems*, 1:335–358, 1987.
17. P. Phillips and I. Lee. Areal aggregated crime reasoning through density tracing. In *Proc. International Workshop on Spatial and Spatio-temporal Data Mining*, 2007.

18. I. Reinbacher, M. van Kreveld, and M. Benkert. Scale dependent definitions of gradient and aspect and their computation. In A. Riedl, W. Kainz, and G. A. Elmes, editors, *Proc. 12th Intern. Symp. Spatial Data Handling (SDH'06)*, pages 863–879, 2006.
19. W. Robinson. Ecological correlations and the behavior of individuals. *American Sociological Reviews*, 15:351–357, 1950.
20. M. Sharir. On k-sets in arrangements of curves and surfaces. *Discrete Comput. Geom.*, 6:593–613, 1991.
21. J. Snow. *On the Mode of Communication of Cholera.* Churchill Livingstone, London, 2nd edition, 1854.