#### CSci 5715, Fall 20: Homework 3

#### Table of Participation

|  |  |  |
| --- | --- | --- |
| Question ID | Answer drafted by | Answer reviewed by |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |

Chapter 5: Query processing and optimization

**Q1**. In public health, contact tracing is the process of identification of persons who may have met an infected person ("contacts") and subsequent collection of further information about these contacts. Diseases for which contact tracing is commonly performed include sexually transmitted infections (including HIV), novel infections (e.g., SARS-CoV, H1N1, and **COVID-19**), etc. In addition, we have learned about the spatial database operation model, namely, *point query, range query, nearest neighbor, and spatial join*. Consider the following scenarios and choose which of the spatial data operation can be used to model individual in the context of contact tracing. Briefly justify your answer.

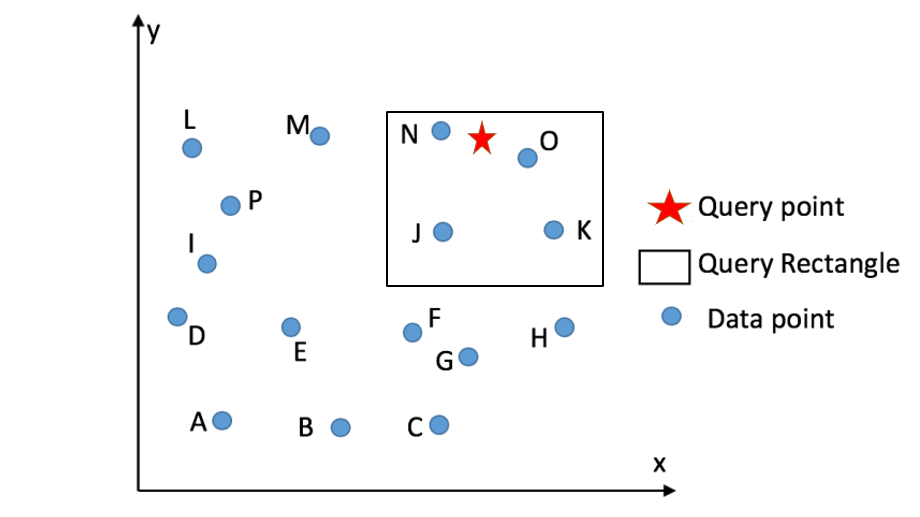
**Q1(a):** A spatial trajectory is a trace generated by a moving object in geographical spaces, usually represented by a series of chronologically ordered points, where each point consists of a geospatial coordinate set and a time stamp. Assume you are given individual GPS trajectory of smart phones that people carry. You are asked to determine the contact of individuals who were within 6 feet in 10 mins time interval based on the given trajectory data sets.

**Ans:** Range Query. With a GPS trajectory table, we aim to find all the contacts that are within 6 ft distance with a specific individual for time stamps given by the time interval.

**Q1(b):** Each mobile device listens to Bluetooth of nearby phones, and then determines the distance based on the power of signal. Assume you are given individual Bluetooth logs of a mobile device and asked to determine the contact based on a distance *h*, and time threshold *T*.

**Ans:** Spatial Join. Here, we use two tables of phone-owner records and bluetooth logs of a specific individual. We aim to find the number of contacts that the bluetooth logged as distance lesser than or equal to h, and the time of contact is greater than or equal to T.

**Q2**. Given 16 points A through P in space shown in Figure 1.1, which are stored in a spatial database, answer the following questions. Assume that one data page can store the information of at most two points.



**Figure 1.1.** 16 points in space (best in color)

**Q2 (a)**. If the points are saved without order, what is the minimum number of data pages need to be retrieved to know:

1. Whether there is a point at the query point?

**Ans: 8 (worst case - linear search needs to visit all data pages)**

1. How many points are there in the query rectangle?

**Ans: 8 (again, with unordered data, requires all data pages)**

1. What is the nearest point of the query point?

**Ans: 8 (same explanation as before)**

**Q2 (b).** Now points are saved in the database using their Z-curve value. The Z-curve used is shown in Figure 1.2 as blue dashed lines. What is the minimum number of data pages need to be retrieved to know?

1- (C,H), 2-(G,E), 3-(A,B), 4-(E,D), 5-(K,J), 6-(O,N), 7-(I,P),8-(M,L)

1. Whether there is a point at the query point?

Z-order value of Query point=12. Using Binary search on data pages,

iteration 1: 4-8,

iteration 2: 6-8,

iteration 3: 6-7,

Hence, number data pages retrieved = 3

1. How many points are there in the query rectangle?

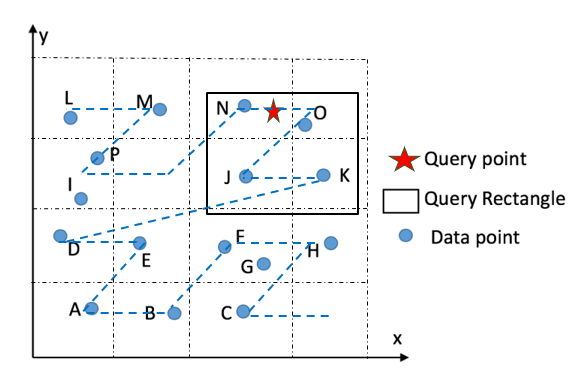
One Z-interval (K-N) or (9-12)

Binary Search to find this interval (page 5) from 8 data pages=3 data pages retrieved. Linear Search from K to N.

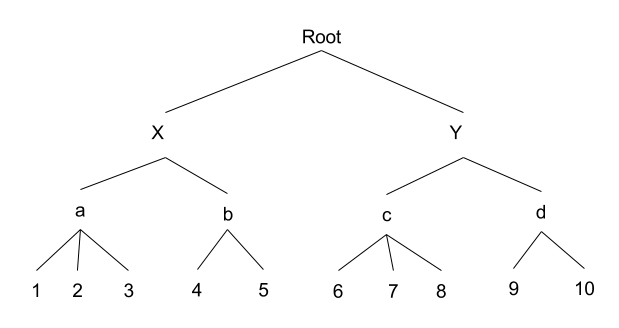
1. What is the nearest point of the query point?

Z-order value of nearest data of query point-12. Hence, a similar binary search query will retrieve 3 data pages.

Briefly explain your reason.



**Figure 1.2.** Points in space with Z-curve grid.

**Q3.** Figure 2(a) shows a distribution of 10 rectangles in dataset RD1 = {1, 2, …, 10} and minimum orthogonal bounding boxes (MBRs, e.g., X, Y, a, b, c, d) in its R-tree index shown in Figure 2(b).

|  |  |
| --- | --- |
|  |  |
| Figure 2(a): Distribution of Rectangles and MBRs | Figure 2(b): R-tree Index |

A nearest-neighbor query is performed to find the nearest data rectangle of the query point **p** represented as a circle in Figure 2(a). Assume that the distance between a point p1 and a rectangle R is the distance between the point p1 and the closest point in the rectangle R. The distances from the query point **p** to the closest and the farthest points in each rectangle and MBR are shown in Table 1.

**Table 1.** Distance from Query Point to Rectangles and MBRs

|  |  |  |
| --- | --- | --- |
| Rectangle/MBR | Closest distance | Farthest distance |
| 1 | 23 | 29 |
| 2 | 22 | 28 |
| 3 | 12 | 18 |
| 4 | 32 | 39 |
| 5 | 39.5 | 45 |
| 6 | 10 | 12.5 |
| 7 | 5 | 12 |
| 8 | 3 | 12 |

|  |  |  |
| --- | --- | --- |
| Rectangle/MBR | Closest distance | Farthest distance |
| 9 | 3 | 9 |
| 10 | 5 | 18 |
| a | 12 | 29 |
| b | 30 | 48 |
| c | 0 | 12.5 |
| d | 3 | 18 |
| X | 12 | 48 |
| Y | 0 | 18 |

**Q (3a)** Show the execution trace of the query following the **Two-phase** Nearest Neighbor algorithm.

Fill out the following table to list the rectangles and MBRs tested in each phase.

|  |  |
| --- | --- |
| Phase | Rectangles/MBRs |
| Phase 1 | Y, c, d, 7, 8, 9, 10 |
| Phase 2 | X, a, 3 |

**Q (3b)** Show the execution trace of the query following the **One-phase** Nearest Neighbor algorithm which recursively checks and eliminate nodes dominated by some other nodes in R-tree index and check the remaining data blocks for nearest neighbor.

Fill out the following table to list the rectangles and MBRs tested in each R-tree level.

|  |  |
| --- | --- |
| Level | Rectangles/MBRs |
| 1st level | X, Y |
| 2nd level | a,~~b~~,c,d |
| Data rectangle | ~~1,2~~,~~3~~, ~~4,5~~, ~~6~~, 7, 8, 9, 10 |

Chapter 7: Spatial Data Mining

**Q4**. Compare two prediction models for COVID-19, namely the black-box and glass-box models. The former fits a curve [1] (e.g., Gaussian or neural network) to the time-series of number of mortalities to predict future mortalities. The latter divides people into four groups (namely, [susceptible, exposed, infected, recovered](https://en.wikipedia.org/wiki/Compartmental_models_in_epidemiology) ) [1] and models transition-processes (e.g., mobility, contact, disease transmissions, hospitalization).

Answer following questions:

1. Which model requires less data?

The **curve-fitting models** typically require less data from fewer sources to make predictions. To develop an SEIR model, more information is needed on the health and habits of the population than for the more basic curve-fitting models. That said, both model types benefit form having more data to fit the model.

1. Which model is more transparent?

The curve-fitting models are generally less transparent than the **SEIR models**, especially when using neural networks which are a classic example of a black-box model. However, both models rely heavily on assumptions about the population, transmission, and public health policies. The assumptions used by the SEIR models are often more clearly understandable than those used by the curve-fitting models.

1. Which model can easily compare impact of interventions such as social distance, stay at home, and compulsory mask wearing?

SEIR models incorporate assumptions about social mixing that curve-fitting models do not explicitly use. This makes it easier for SEIR models to compare expected impacts of interventions.

1. Which model's conclusions suffer more from non-stationarity introduced by policy interventions such as stay at home or business reopening?

If a sudden policy intervention is introduced at some point during the pandemic, the curve-fitting model predictions will likely suffer more. This is related to the answer to question iii. The influence of previous data (and assumptions about the nature of the data) on the model prediction will likely fail to properly account for sudden behavioral changes in the population caused by policy interventions.

The conclusions of SEIR models are generally more sensitive to changes in social mixing caused by policy interventions because they model the disease transition processes. Because of this sensitivity, makers of SEIR models can anticipate the effects of policy changes before they happen by assessing the effects of changing model parameters (e.g. the effects of contact reduction in the Columbia University Severe Covid-19 Risk Model).

Still, both models are at least partially able to account for changes in public health policies by altering the assumptions they use.

**Q5**. Given a distribution for a total of ~~19~~ 21 points shown in Figure 3.1, answer the following questions.

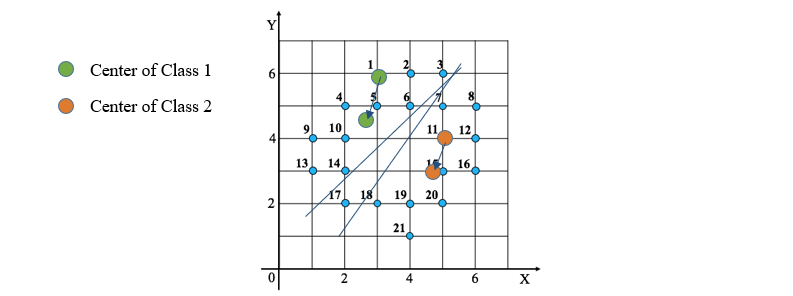


Figure 3.1. Points distribution

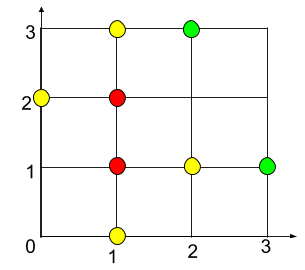
**Q5 (a)**. Trace execution of K-Means algorithm with for the points in Figure 3.1 and fill out the following table with results in each iteration. Assume that the seeds are (5,4) and (3,6). (The answers might not fill all the rows of the empty tables.) **Note.** If there are some points that are equi distant from cluster center, you can place the points in any of the two cluster.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Iteration ID | Points ∈ Class-1 | Center of Class 1 | Points ∈ Class-2 | Center of Class 2 |
| 1 | 7, 8, 11, 12, 15, 16, 17, 18, 19, 20, 21 | (5,4) | 1, 2, 3, 4, 5, 6, 9, 10, 13, 14 | (3,6) |
| 2 | 7, 8, 11, 12, 15, 16, 18, 19, 20, 21 | (4.64, 3) | 1, 2, 3, 4, 5, 6, 9, 10, 13, 14, 17 | (2.7, 4.7) |
| 3 | 7, 8, 11, 12, 15, 16, 18, 19, 20, 21 | (4.9, 3.1) | 1, 2, 3, 4, 5, 6, 9, 10, 13, 14, 17 | (2.64, 4.45) |
| 4 | Return |  | Return |  |
| 5 |  |  |  |  |
| 6 |  |  |  |  |
| 7 |  |  |  |  |
| 8 |  |  |  |  |

**Q5 (b)**. Density-based spatial clustering of applications with noise (DBSCAN) is another data clustering algorithm. For the purpose of DBSCAN clustering, the points to be clustered are classified as *core points*, (*density*-)*reachable points* and *outliers*, as follows:

* A point *p* is a core point if at least  points are within distance *ε* (*ε* is the maximum radius of the neighborhood from *p*) of it (**including *p***). Those points are said to be *directly reachable* from *p*. No points are *directly reachable* from a non-core point.
* A point *q* is reachable from *p* if there is a path *p*1, ..., *pn* with *p*1 = *p* and *pn* = *q*, where each *pi*+1 is directly reachable from *pi* (all the points on the path must be core points, with the possible exception of *q*).
* All points not reachable from any other point are outliers.

For example, as shown in Figure 3.2 if and , red points are core points, and the yellow ones are density-reachable from them, and outliers are represented in green.

  
Figure 3.2. A DBSCAN example

Use DBSCAN to cluster the points in Figure 3.1 and fill out Table 1 and 2 with information of the final clusters and outliers. **4** and (The answers might not fill all the rows of the empty tables.)

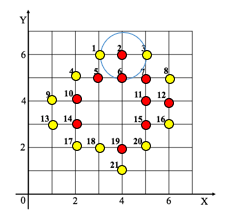
Table 2. Clusters

|  |  |  |
| --- | --- | --- |
| Cluster # | Density-reachable points | Core points |
| 1 | 18, 20, 21 | 19 |
| 2 | 4, 9, 13, 17 | 10, 14 |
| 3 | 1, 3, 8, 16 | 2, 5, 6, 7, 11, 12, 15 |
|  |  |  |
|  |  |  |

Table 3. Outliers (Leave it blank if there is no outliers)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Point # |  |  |  |  |  |  |  |

**Note: No outliers found for the dataset from Figure 3.1. See below.**

****

**Q5 (c)**. In order to detect hotspots with high log likelihood ratio (LogLR), we check the LogLR of four circular areas shown in Figure 3.3 which are potential hotspots. LogLR is calculated as follows.

where c is the number of points (blue dots ) inside a candidate region, ctot is total number of points in the study area, e is the expectation of number of points inside the candidate region, areac is the area of the candidate region and areatot is that of the whole region.

**Note:** Study area is the surface area of the graph. Here it is 7\*7 = 49 units. Similarly, for A it would be the surface area of the circle (Pi r ^2).

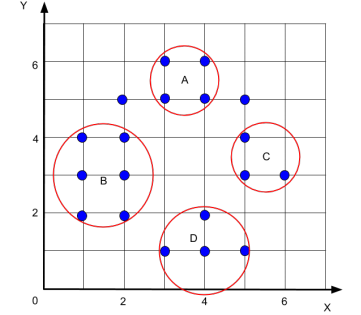


Figure 3.3. Points with circular areas

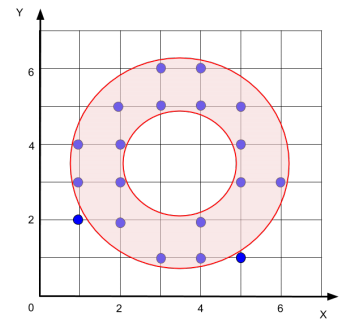


Figure 3.4. Points with a donut-shape area

Table 4: Information on areas

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute | Whole region | A | B | C | D |
| Area | 49 | 1.6 | 4 | 1.6 | 3.8 |

Fill out Table 5 to show the results of log likelihood ratio computation and select one candidate region to be the most likely hotspot.

Table 5. Likelihood ratio results

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | A | B | C | D |
| c | 4 | 6 | 3 | 4 |
| e | 0.62 | 1.55 | 0.62 | 1.47 |
| ctot - c | 15 | 13 | 16 | 15 |
| ctot - e | 18.38 | 17.45 | 18.38 | 17.53 |
| LogLR | 1.91 | 1.86 | 1.09 | 0.72 |

**Q5 (d)**. After learning the nature of these point records, we propose that maybe donut-shape areas are more suitable for delineating hotspots in this scenario. Compute the LogLR of the area shown in Figure 3.4 and test this hypothesis. The area of the donut-shape region is 12.5.

For the donut shape, the LogLR can be computed as follows.

|  |  |
| --- | --- |
|  | Donut |
| c | 17 |
| e | 4.85 |
| ctot - c | 2 |
| ctot - e | 14.15 |
| LogLR | 7.57 |

This seems to confirm the alternate hypothesis that the donut-shape area is more suitable for this scenario.

**Q6.** Consider 9 cells, namely, A, B, …, I, in a raster dataset. The location and attribute values of these cells are shown in Figure 4. Assume that the neighborhood of a cell consists of the cells sharing an edge with the cell. For example, B and D are neighbors of ~~C~~ A.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | A | B | C | | D | E | F | | G | H | I | | |  |  |  | | --- | --- | --- | | 42 | 39 | 37 | | 38 | 36 | 34 | | 36 | 34 | 41 | |
| **Figure 4(a).** Location of Cells | **Figure 4(b).** Attribute Values of Cells |

**Q6a)** Consider spatial Z-test, a quantitative test for spatial outliers. Fill out the following table to prepare the calculation of the values of spatial test .

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E | F | G | H | I |
|  | 38.5 | 38.33 | 36.5 | 38 | 36.25 | 38 | 36 | 37.67 | 34 |
|  | 3.5 | 0.67 | 0.5 | 0 | -0.25 | -4 | 0 | -3.67 | 7 |
|  | 0.92 | 0.075 | 0.025 | 0.12 | 0.20 | 1.32 | 0.12 | 1.22 | 1.96 |

Note that and are data cells, is the neighborhood of , is the attribute value of cells , and is the deviation of ’s attribute value from its neighborhood mean value.

Recall that , where is the mean of , and is the standard deviation of which is ~~3.16,~~  3.35 in this case.   
  
i) What is the value of ?

The mean of is = 0.42.

ii) Assume the threshold to determine an outlier is . List the cells which are spatial outliers.

In this case, cell I is the only spatial outlier with a spatial Z-score of = 1.96, which is above the threshold.

**Q6b)** What is the worst-case asymptotic time complexity of the spatial Z-test to identify spatial outliers given *n* cells? Assume that the neighborhood size is bounded by a constant, e.g., 4.

Given n cells of a raster dataset like the one in Figure 4 where the neighborhood size is bounded (less than or equal to 4), the Z-scores can be computed (and checked against a threshold) in constant time. The time complexity is O(n).

1. Compute mean of neighbors: O(4n) = O(n)
2. Determine S(x): O(n)
3. Compute Z\_s (x): O(n)

**Q6c)** Recall the Variogram cloud is a graphical test for spatial outliers. What is the worst-case asymptotic time complexity of Variogram cloud when the input raster has *n* cells?

The Variogram cloud requires computing the square root of the absolute difference between each pair of values and the pairwise distances. For a raster with n cells, there are n(n-1)/2 pairs of cells. Thus, the Variogram cloud values can be computed in quadratic time. The time complexity is O(n^2).

References:

[1] COVID-19 Models: Can They Tell Us What We Want to Know? Michaud Josh, Kates Jennifer, and Levitt Larry, April 2020, <https://www.kff.org/policy-watch/covid-19-models/>