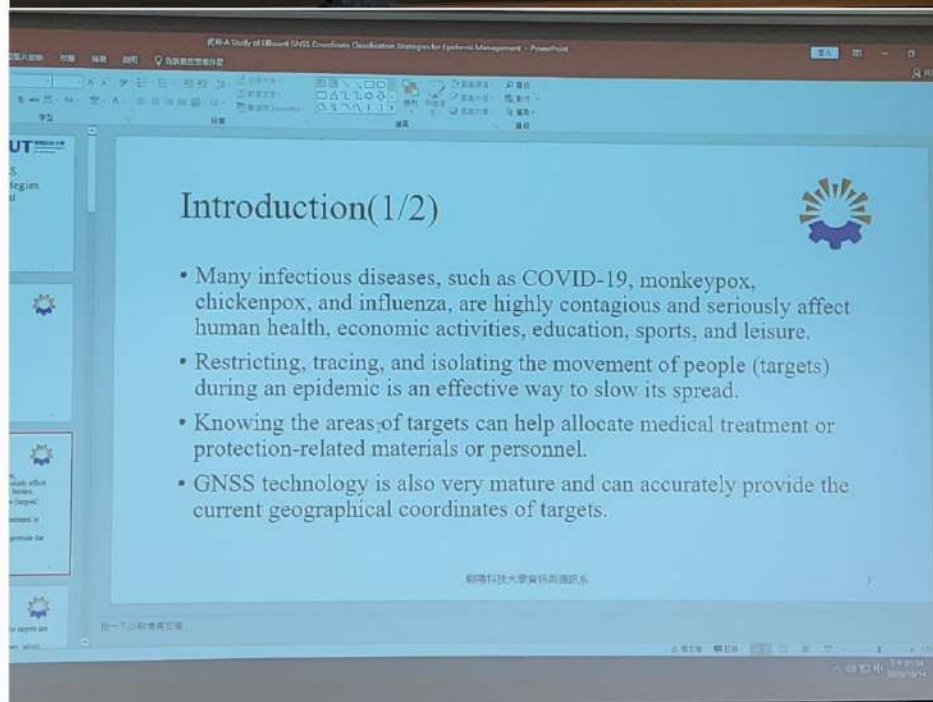
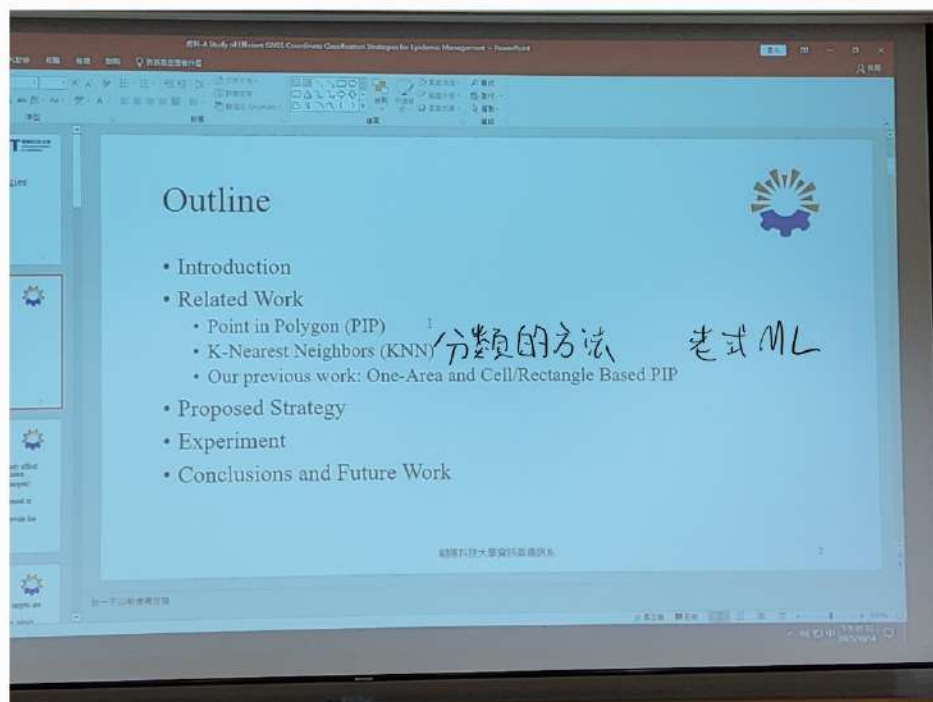
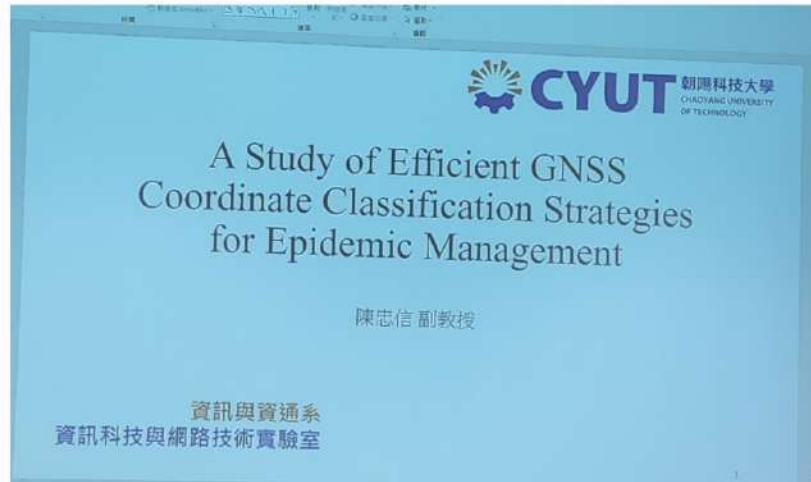


日期: 2025/10/14

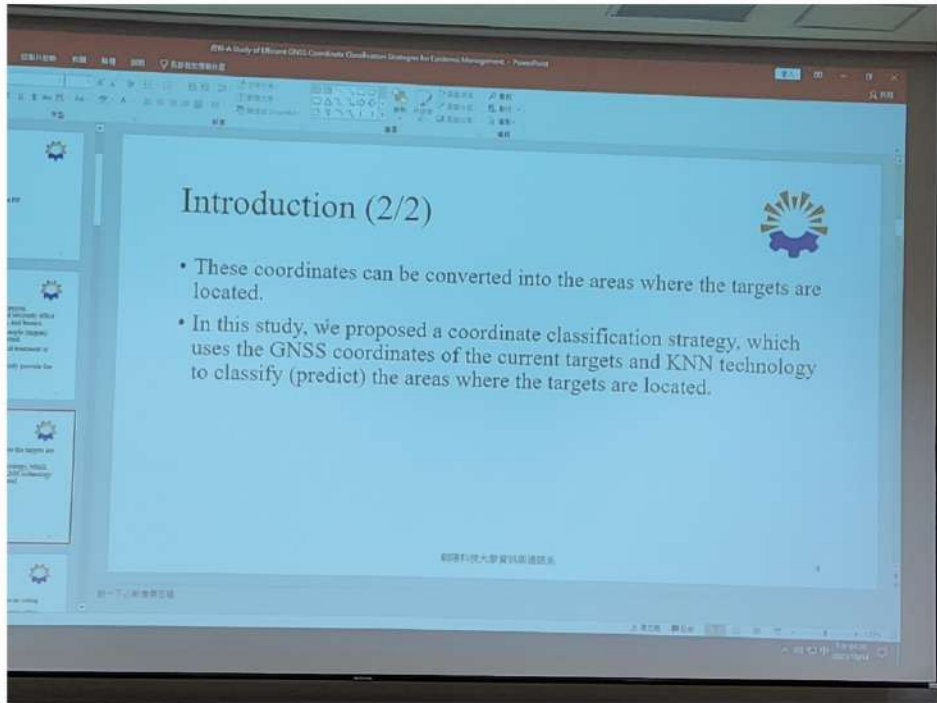
講者: 陳忠信

A Study of Efficient GNSS Coordinate Classification Strategies for Epidemic Management

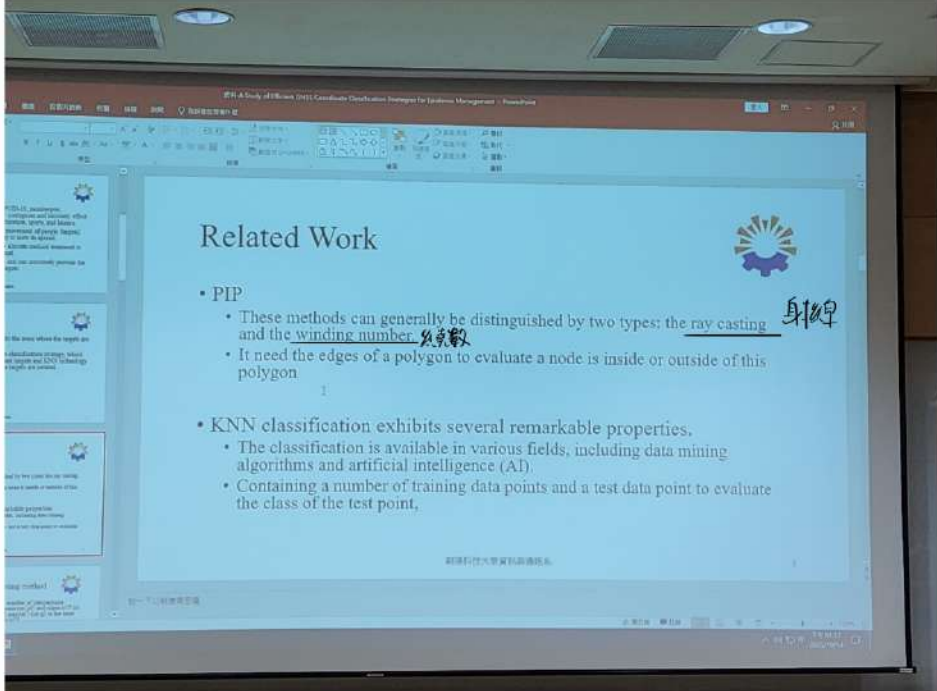


GNSS已經足夠成熟
能回傳目標精確的座標

GNSS座標可以轉換成
所屬區域
以目標當前的座標+kNN
分類-預測在哪个區域

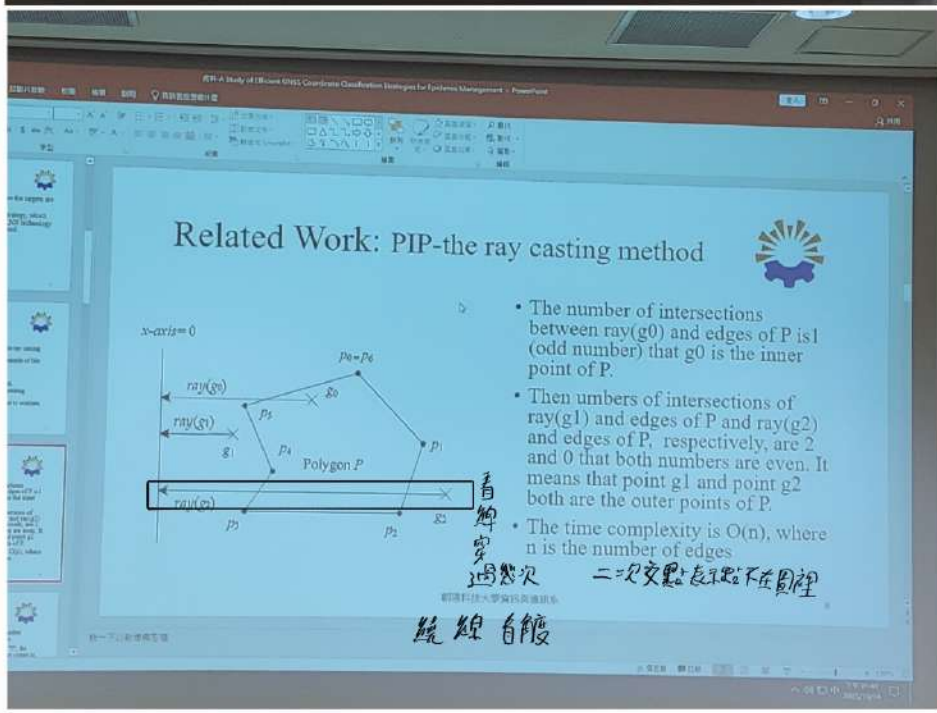


PIP (Point-in-Polygon)
射線法
判斷點是否在多邊形裡
kNN分類
用於資料標勘與AI

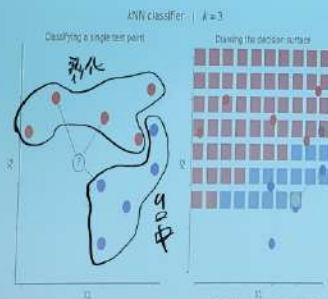


與多邊形交點
odd -> inside
even -> outside

時間複雜度 $O(n)$ (n為邊數)



Related Work: KNN



先全掃描
再看最近的點
投票

- Application of a k -NN classifier considering $k = 3$ neighbors.
- Left - Given the test point "?", the algorithm seeks the 3 closest points in the training set, and adopts the majority vote to classify it as "class red".
- By iteratively repeating the prediction over the whole feature space (X_1, X_2), one can depict the "decision surface".

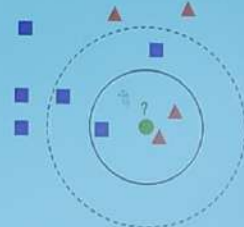
KNN → 資料數值化



Related Work: Traditional KNN

- In the KNN classification, three steps are involved for a test data point as follows:
 - Step 1. It evaluates the Euclidean distance between the sample and the test point, with a time complexity of $O(n \cdot D)$, where n is the data set size.
 - Step 2. It sorts the training dataset based on Euclidean distances with an $O(n \cdot D^2)$ time complexity.
 - Step 3. It uses the majority classification rule to predict the class of the test point, with a time complexity of $O(k)$, where k is the number of neighbors in the KNN classification.

For example $k=3$



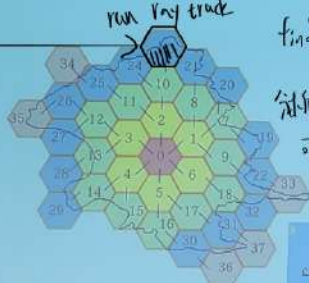
計算每個點在測試點距離

sort
find 3 doc
vote

$O(n^2)$
 $n \rightarrow$ train data set



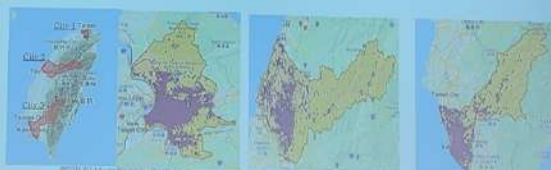
Related Work: One-Area and Cell/Rectangle Based PIP



效果比較好
find how many topic inside

split cell into
可知是在 polygon 之外

- Cell Allocation for a target area.
- Techniques:
 - Points in inner cells: unnecessary do evaluation
 - Points in intersected cells: do evaluation



Step 1: 計算測試點到所有
樣本的歐氏距離 $O(n \cdot D)$

Step 2: sort $O(n \cdot D^2)$

Step 3: 取前 k 個做多數決
 $O(k)$

對目標區做分割 加速判斷

模糊的區塊做 PIP

Related Work: One-Area and Cell/Rectangle Based PIP



.. Distribution of points in hot spot one with 6024 points.

把範圍區域切成矩形

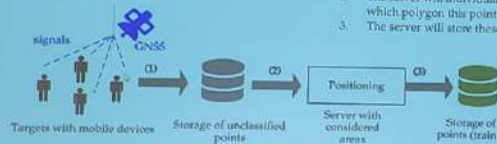
- (a) Distribution of spots and (b) distribution of highways.



System Model

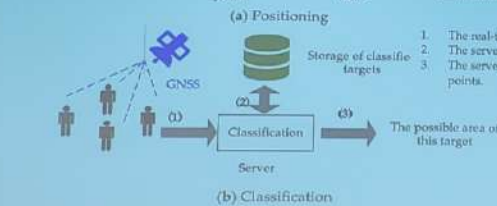


1. The data sent from the mobile devices of targets will be stored as unclassified geographical points.
2. The server will individually take out a point from the unclassified points and position which polygon this point is inside.
3. The server will store these geographical points with their classes.



store point → classification

先有



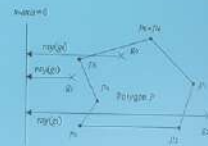
1. The real-time data sent from targets' mobile devices arrive at the server.
2. The server extracts candidate points from the storage of classified targets.
3. The server will classify the targets into their located areas according to the candidate points.

Proposed Strategies: PIP Implementation



線交點 g_0 為 P_1 P_2

Algorithm 1: SegSegInter (point g_0 , point g_1 , point g_2 , point g_3)
 Input: g_0, g_1, g_2, g_3 are points that form a segment from g_0 to g_1 and a segment from g_2 to g_3 .
 Output: The result is 1 or 0, indicating which two line segments intersect or do not.
 Method: Algorithm for evaluating the intersection of two segments
 1. $res = 0$;
 2. $n = (g_1x - g_0x) \times (g_2y - g_0y) - (g_1y - g_0y) \times (g_2x - g_0x)$;
 3. if $n \neq 0$ then
 4. $res = 0$;
 5. else
 6. $x1 = (g_1x - g_0x) \times (g_2y - g_0y) - (g_1y - g_0y) \times (g_2x - g_0x)$;
 7. $x2 = (g_1x - g_0x) \times (g_3y - g_0y) - (g_1y - g_0y) \times (g_3x - g_0x)$;
 8. if $(x1 \geq 0 \text{ and } x1 \leq 1) \text{ and } (x2 \geq 0 \text{ and } x2 \leq 1)$
 9. then
 10. $res = 1$;
 11. else
 12. $res = 0$;
 13. end if
 14. end if
 15. output res



Segment from g_0 to g_1 is one edge of this polygon.
 Ray(g_0): segment from g_0 (g_0_x, g_0_y) to ($0, g_0_y$)

定位

儲存座標

PIP判定區域

存入 Database (Mark and cmin)

分類

座標傳進 Database

從已分類 Database 取候選點

KNN 將目標分類到對應區域

計算是否相交進而完成 PIP

Proposed Strategies: PIP Implementation



Algorithm 2: PIP_EP (point g , polygon P)
 Input: g is a point and $P = \{p_1, p_2, \dots, p_n\}$ is a polygon.
 Output: The result is 1 or 0, indicating that g is located in P or not.
 Method: /an algorithm for positioning a point to a polygon
 1. $count := 0$; $result := 0$;
 2. for $i := 0$ to $n - 1$ do //each edge of polygon P
 3. $g'_x := 0$; $g'_y := 0$;
 4. If $SegSigned(p_i, p_{i+1}, g, g') = 1$ then
 5. $count := count + 1$;
 6. end if
 7. end for
 8. if $(count \% 2 = 1)$ then $result := 1$;
 9. else $result := 0$;
 10. end if
 11. output $result$.

According to Algorithms 1, 2, and 3, PIP is implemented.

Algorithm 3: PIPes (point g , polygon set P_A)
 Input: g is a point and $P_A = \{P_0, P_1, \dots, P_{m-1}\}$ is a polygon set.
 Output: The result is a value i indicating that point g is located in polygon P_i , where $0 \leq i \leq m - 1$.
 Method: an algorithm for positioning the located polygon of point g .
 1. for $i := 0$ to $m - 1$ do
 2. if $PIPE(g, P_i) = 1$ then
 3. break;
 4. end if
 5. end for
 6. output i .

交點 \rightarrow even \rightarrow point not in polygon
 \rightarrow odd \rightarrow point in polygon

KNN 允許計算相近點的類別
 取最多為結果
 加入權重 (距離越近權重越高)
 由多數決改進為距離加權

Proposed Strategy: KNN Classification Implementation



- For KNN classification will make statistics on the class g'_{pc} of g' using $I(-)$, find the class i with the largest number, and then assign it to g_{cc} .

$$g_{cc} = \arg(\max_i \sum_{g' \in NB} I(g'_{pc} = i))$$

- We also contain the weighting KNN
- The Euclidean distance of two points ga and gb is as follow, where (ga_x, ga_y) is the coordinate value of point ga and (gb_x, gb_y) is the coordinate value of point gb .

$$d(ga, gb) = \sqrt{(ga_x - gb_x)^2 + (ga_y - gb_y)^2}$$

- The weighting KNN is

$$g_{cc} = \arg(\max_i \sum_{g' \in NB} I(g'_{pc} = i) \times d(g, g')^{-1})$$

權重
weight
Distance 可以計算距離

Adaptive KNN

搜尋鄰近點
 給每個點加權
 sort後統計權重
 選出權重最高的類別

搜尋全部資料



當找到不夠
就擴大

Step 2 weight

Algorithm 4: AdaptiveKNN (point g , numerical value r , training dataset T , P_A , integer k)
 Input: g is a point, r is a numerical value, T is a training dataset, and k is a specific number.
 Output: the class of point g .
 Method: /an algorithm for adaptive KNN classification
 Notation and Initialization:
 - m : the size of polygon set P_A
 - NB : a set for storing the neighbors of point g , where the arrangement of NB is $(NB(0), NB(1), \dots, NB(m-1))$
 - V : a numerical list $(V(0), V(1), \dots, V(m-1))$ for a vote. The initial value of $V(i)$ is 0 for $0 \leq i \leq m - 1$.
 1. $r' := 0$; $nb := 0$;
 2. while $nb < k$ do //Step 1: a search of nb-neighbors/
 3. $r' := r' + r$; $nb := nb + 1$;
 4. for each g' in T do
 5. if $g_x \geq (g'_x - 0.5r')$ and $g_x \leq (g'_x + 0.5r')$ and $g_y \geq (g'_y - 0.5r')$ and $g_y \leq (g'_y + 0.5r')$ then
 6. $NB(nb) := g'$; $nb := nb + 1$;
 7. end if
 8. end for
 9. end while
 10. for $i := 0$ to $nb - 1$ do //Step 2: an assign of weight to each neighbor $NB(i)$ /
 11. $NB(i)_{w_i} := 1 / d(g, NB(i))$;
 12. end for
 13. for $i := 0$ to $nb - 2$ do //Step 3: a sort of NB */
 14. for $j := i + 1$ to $nb - 1$ do
 15. if $NB(i)_{w_i} > NB(j)_{w_j}$ then //Swapping of $NB(i)$ and $NB(j) + 1$
 16. $g' := NB(i)$; $NB(i) := NB(j) + 1$; $NB(j + 1) := g'$;
 17. end if
 18. end for
 19. end for
 20. //Step 4: a classification for point g /
 21. for $i := 0$ to $i = 1$ do //Step 4-1: An accumulation of weight for the class $NB(i)_{w_i}$ /
 22. $i := NB(i)_{w_i}$; $V(i) := V(i) + NB(i)_{w_i}$;
 23. end for
 24. $g_{cc} := 0$ //Step 4-2: A search of class g_{cc} that the accumulation of weight is largest */
 25. for $i := 1$ to $v_{i-1} = 1$ do
 26. if $V(i) > V(g_{cc})$ then
 27. $g_{cc} := i$;
 28. end if
 29. end for
 30. output g_{cc} .

改良的KNN不需要搜尋全部的資料能縮短分類時間

Proposed Strategy: KNN Classification Implementation

- Algorithm 4 employs the technology of the weighting KNN classification for classifying points into areas.
- In addition, Step 1 of this algorithm calculates the candidates of k neighbors based on a numerical value r . When necessary, the value of r will be adaptively adjusted until the number of candidates is greater than or equal to k .
- So, the candidates of k neighbors in Steps 2, 3, and 4 are k or slightly more than k data points, not the total training dataset.
- In this way, we improve the classification time

19

PIP $O(m \times n_{max})$
 m : polygon 數量
 n_{max} : 最大邊數

Proposed Strategy: Analysis

Property 1. Given a point g and a polygon set P_A with size m , if point g is inside one of set P_A , Algorithm PiPos positions point g in $O(m \times n_{max})$ time, where n_{max} is this polygon's largest edge number of this polygon set.

Property 2. Given a point g and a training dataset T with size n_T , algorithm AdaptKNN classifies point g in $O(n_T)$ time.


20

AdaptKNN $O(n_T)$
 n_T : 訓練集大小

台北市 實驗點
打卡點當一個人

Experiment: Environment

- The experimental environment consists of the scope of a geographic area and a set of geographic points within the area.
- The area is a famous city, ranging from 120.6 to 122.9 east longitude and 24.8 to 25.4 north latitude.
- The distribution of data points.



21

改良KNN
in Big Data

Conclusions



- In this paper, we have planned a strategy, including positioning and classification phases, which can be used when epidemic management or other applications need to track the location of some targets or people.
- We hope this research can help epidemic management understand the spread of these pathogens and enable us to make predictions and preparations earlier, significantly as the infection numbers rapidly increase.