

A Study of Efficient GNSS Coordinate Classification Strategies for Epidemic Management

日期：2025/10/14

講者：陳忠信 副教授

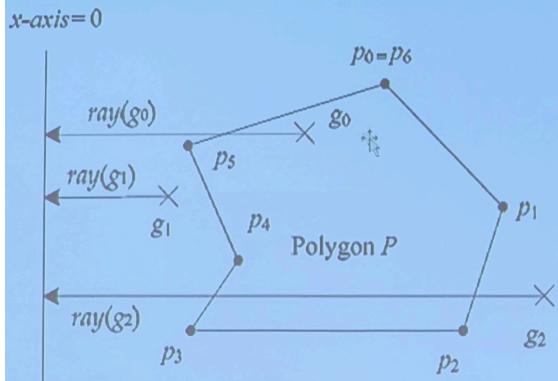
Introduction

- GNSS 高準確性
 - 誤差幾公尺

Point in Polygon (PIP)

- 判斷點是否在多邊形內
 - 射線：交點問題
 - 穿過幾次來判斷是否在多邊形內
 - 繞線：角度問題
 - 以邊為準，設定：
 - 幾度是進
 - 幾度是出

Related Work: PIP-the ray casting method



- The number of intersections between ray(g_0) and edges of P is 1 (odd number) that g_0 is the inner point of P .
- Then numbers of intersections of ray(g_1) and edges of P and ray(g_2) and edges of P , respectively, are 2 and 0 that both numbers are even. It means that point g_1 and point g_2 both are the outer points of P .
- The time complexity is $O(n)$, where n is the number of edges

K-Nearest Neighbors (KNN)

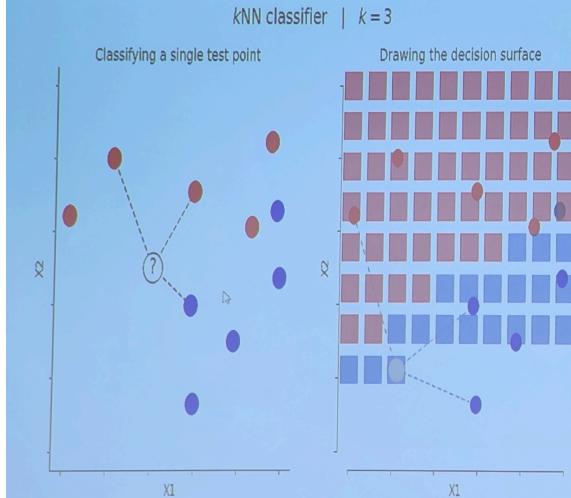


蒐集很多標籤 → 分類
貼完標籤 → 投票預測

1. 圓點 → 已分類好的點
2. 全部掃描完 → 找最接近的點



Related Work: KNN



- 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".

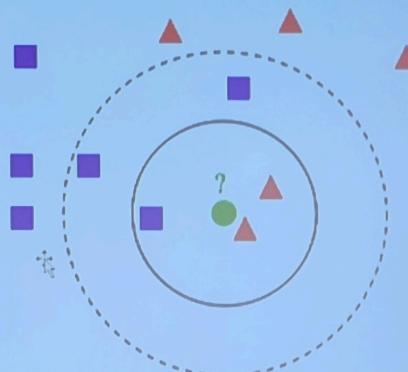
- 資料要數值化
- Step 2 針對大資料量

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 $O(nTD)$, where nTD is the data set size.
 - Step 2. It sorts the training dataset based on Euclidean distances with an $O(nTD^2)$ 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.

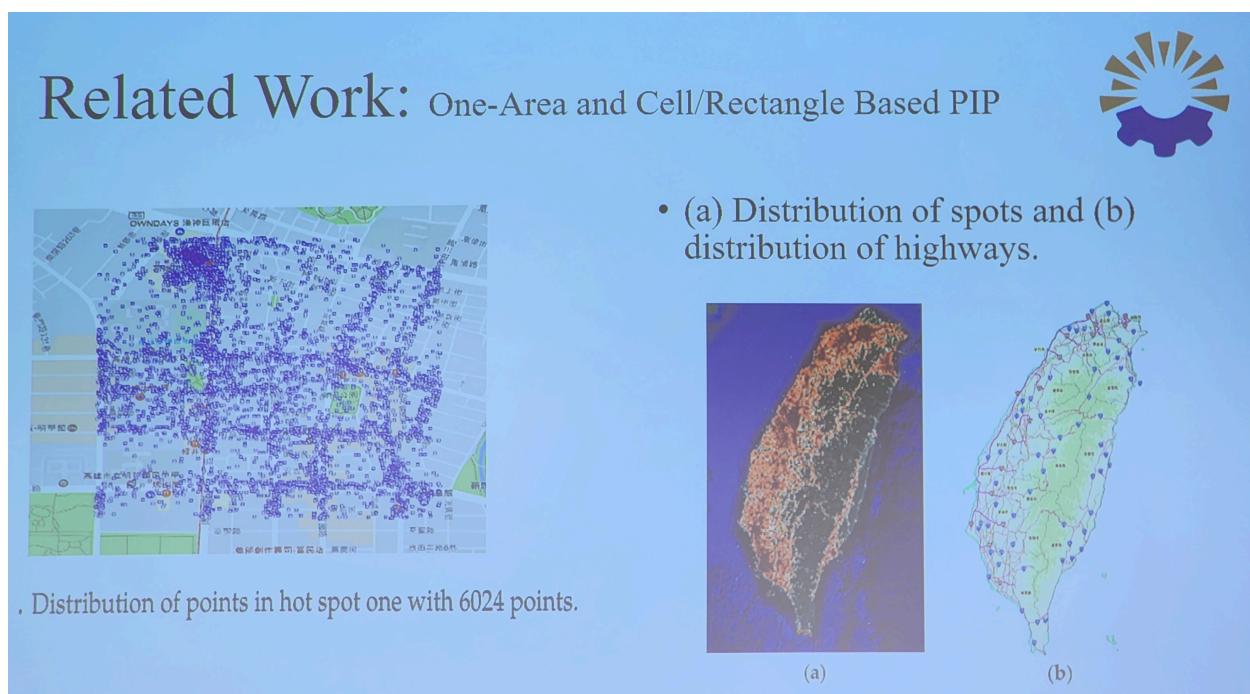
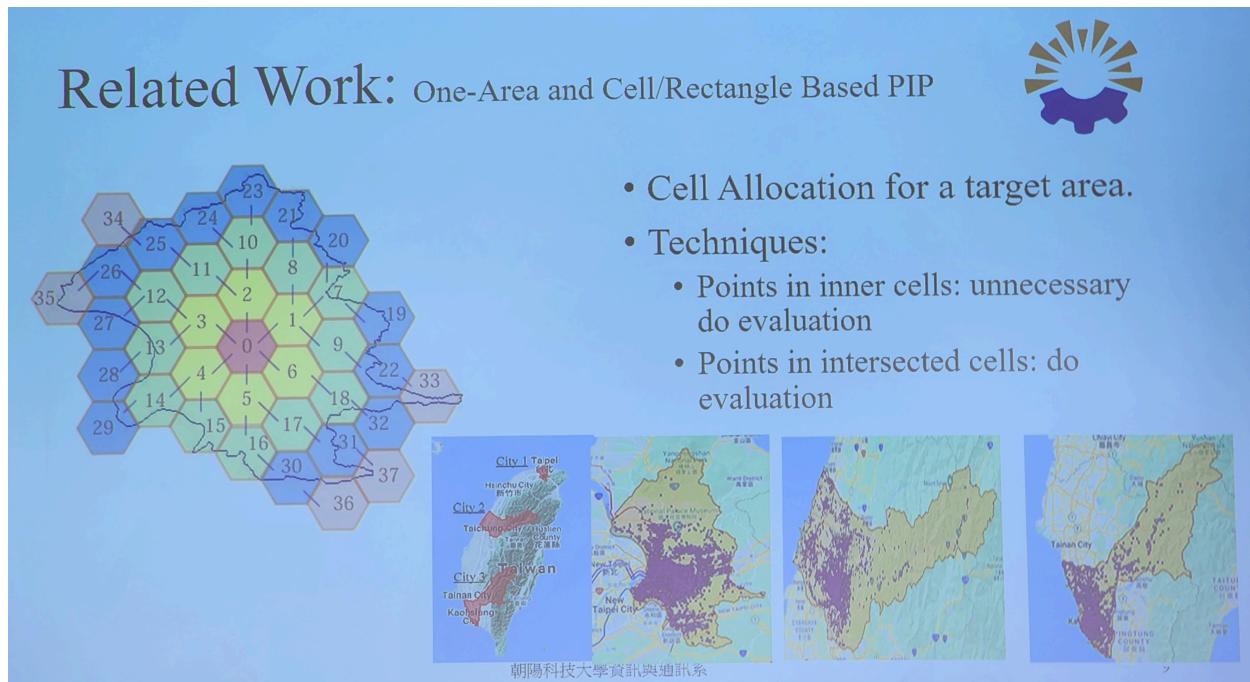
- For example $k=3$



Source https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

One-Area and Cell/Rectangle Based PIP

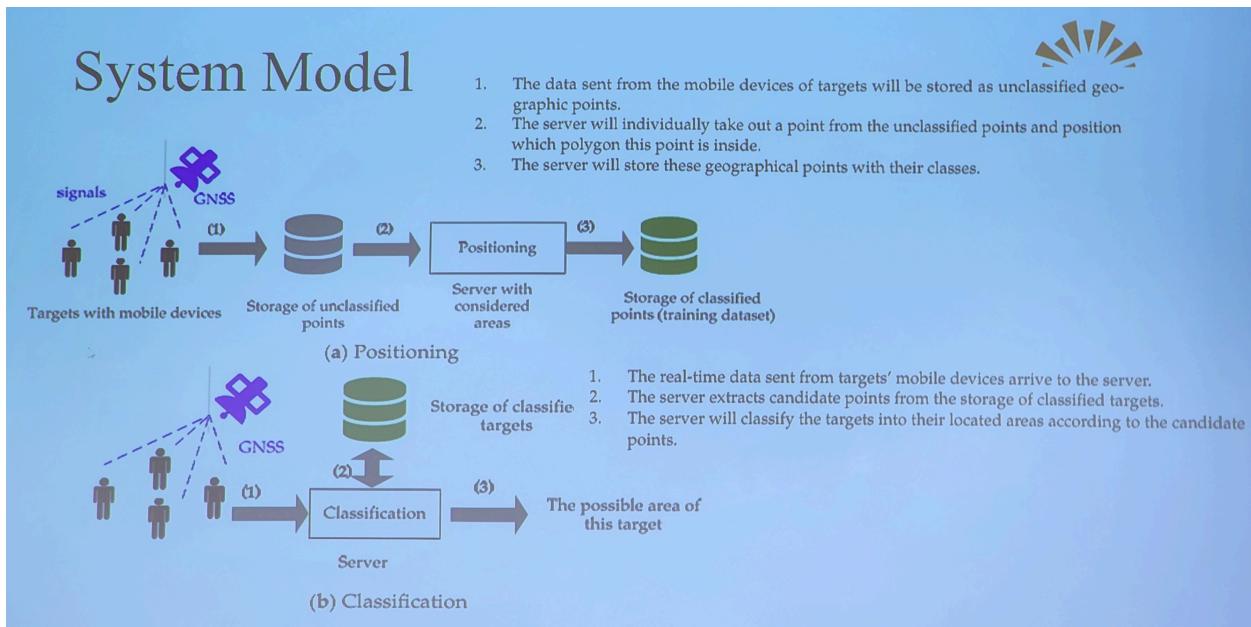
- 用CELL配置區域
 - 無線網路切CELL 代表基地台的位置
- 有交界才要跑PIP
- 資料會重複 → 改正方形的配置



System Model



手機 gps座標 → 定位 → 上標籤 → 訓練 → 分類



Strategy

- 段落跟段落有沒有交點
 - 射線跟邊有沒有交接
- 計算斜率
 - 斜率 0
 - 沒有交點
 - 斜率 其他
 - 判斷交點位置
- 交點

- 奇數 → 裡面
- 偶數 → 外面

Proposed Strategies: PIP Implementation



Algorithm 1: SegSegInt (point ga, point gb, point gc, point gd).

Input: ga, gb, gc, gd are points that form a segment from ga to gb and a segment from gc to gd
Output: The result is 1 or 0, indicating which two line segments intersect or do not.

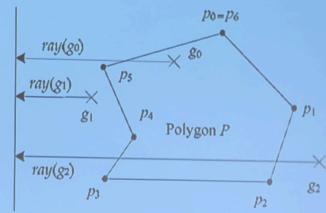
Method://an algorithm for evaluating the intersection of two segments

```

1.   result := 0;
2.    $\alpha := (ga_x - gb_x) \times (gc_y - gd_y) - (ga_y - gb_y) \times (gc_x - gd_x);$ 
3.   if  $\alpha = 0$  then
4.     result := 0;
5.   else
6.      $\kappa_1 := ((ga_x - gc_x) \times (gc_y - gd_y) - (ga_y - gc_y) \times (gc_x - gd_x)) / \alpha;$ 
7.      $\kappa_2 := ((ga_x - gb_x) \times (ga_y - gc_y) - (ga_y - gb_y) \times (ga_x - gb_x)) / \alpha;$ 
8.     if ( $\kappa_1 \geq 0$  and  $\kappa_1 \leq 1$ ) and ( $\kappa_2 \geq 0$  and  $\kappa_2 \leq 1$ )
9.       then
10.         result := 1;
11.       else
12.         result := 0;
13.       end if
14.     end if
15.   output result.

```

x-axis=0



Segment from ga to gb is one edge of this polygon
Ray(g0): segment from g0 (g0_x, g0_y) to (0, g0_y)

Proposed Strategies: PIP Implementation



Algorithm 2: PIP_EP (point g, polygon P).

Input: g is a point and $P = \{p_0, p_1, \dots, p_n\}$ 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.      $g'_x := 0; g'_y := g_y;$ 
4.     if SegSegInt( $(p_i, p_{i+1}, g, g')$ ) = 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: PtPos (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 PtInPy(g,  $P_i$ ) = 1 then
3.       break;
4.     end if
5.   end for
6.   output i.

```

- gcc 為預測的分類

- 權重

- 越遠越高

- 越近越低

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} .

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

- We also contain the weighting KNN
- The Euclidean distance of two points ga and gb is as follow, where (gax, gay) is the coordinate value of point ga and (gbx, gby) is the coordinate value of point gb .

$$d(ga, gb) = \sqrt{(gax - gbx)^2 + (gay - gby)^2}$$

- The weighting KNN is

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

Algorithm 4: AdaptKNN (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:

```

1.   m: the size of polygon set  $P_A$ 
2.   NB: a set for storing the neighbors of point  $g$ , where the arrangement of  $NB$  is  $((NB(0), NB(1), \dots))$ 
3.   V: a numerical list  $(V(0), V(1), \dots, V(m - 1))$  for a vote. The initial value of  $V(i)$  is 0 for  $0 \leq i \leq m - 1$ 
4.   r' := 0; nb := 0;
5.   while  $nb < k$  do /*Step 1: a search of  $nb$  neighbors*/
6.     r' := r' + r; nb := 0;
7.     for each  $g'$  in  $T$  do
8.       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
9.         NB(nb) :=  $g'$ ; nb := nb + 1;
10.    end if
11.   end for
12. end while
13. for i := 0 to  $nb - 1$  do/* Step 2: an assign of weight to each neighbor  $NB(i)$  */
14.   NB(i)w := 1/d( $g$ ,  $NB(i)$ );
15. end for
16. for i := 0 to  $nb - 2$  do /* Step 3: a sort of  $NB$  */
17.   for j := i + 1 to  $nb - 1$  do
18.     if  $NB(j)_w > NB(j + 1)_w$  then //Swapping of  $NB(j)$  and  $NB(j + 1)$ 
19.       g' :=  $NB(j)$ ; NB(j) :=  $NB(j + 1)$ ; NB(j + 1) := g';
20.     end if
21.   end for
22. end for
23. /* Step 4: a classification for point  $g$ /*
24. for i := 0 to  $k - 1$  do/* Step 4-1: An accumulation of weight for the class  $NB(i)_{pc}$  */
25.   t :=  $NB(i)_{pc}$ ; V(i) := V(i) + NB(i)w;
26. end for
27. gcc := 0/* Step 4-2: A search of class  $g_{cc}$  that the accumulation of weight is largest */
28. for i := 1 to  $n_{PA} - 1$  do
29.   if  $V(i) > V(g_{cc})$  then
30.     gcc := i;
31.   end if
32. end for
33. output  $g_{cc}$ .

```

朝陽科技大學資訊與通訊系



- 傳統的做法
 - 求距離 → 排序
- 現在的做法
- 只看固定區域
 - 自適應調整區域
 - 直到人數夠 → 加權重

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



降低時間複雜度

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 PtPos 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.

Experiment

- 地點
 - 台北市
- 打卡的地方當作人

Experiment: Environment



The map displays the geographic area of New Taipei City, including districts like Nugu District, New Taipei, and Xindian District. A dense cluster of purple dots represents data points, primarily concentrated in the central urban area around the National Palace Museum and Yangmingshan National Park.

- 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.

- Type 1
 - 切比較大範圍
 - 12 classes
- Type 2
 - 切更細
 - 256 classes

Experiment: Class distributions



- The class distributions of Type-1 and Type-2. Type-1 has 12 classes and Type-2 has 256 classes.



(a) Type-1



(b) Type-2



準確度跟KNN, wKNN 和 awKNN 比較

- awKNN 不保證是最接近的節點

Experiment: Classification Time



Table 2. Classification time based on different sizes in Type-1, where size is the average number of training data points per class.

Classification \ Size	2	4	8	16	32	64	128
knn	5.24×10^{-6}	1.42×10^{-5}	4.36×10^{-5}	1.52×10^{-4}	5.80×10^{-4}	2.30×10^{-3}	9.43×10^{-3}
wknn	6.66×10^{-6}	1.57×10^{-5}	5.17×10^{-5}	1.61×10^{-4}	5.95×10^{-4}	2.41×10^{-3}	9.90×10^{-3}
awknn	3.08×10^{-6}	5.42×10^{-6}	6.25×10^{-6}	1.27×10^{-5}	1.96×10^{-5}	5.07×10^{-5}	1.79×10^{-4}

Note: unit—seconds. Page: 13

Table 3. Classification time based on different size in Type-2.

Classification \ Size	2	4	8	16	32	64	128
knn	3.17×10^{-3}	1.34×10^{-2}	6.38×10^{-2}	2.74×10^{-1}	1.12×10^0	4.91×10^0	2.29×10^1
wknn	3.21×10^{-3}	1.37×10^{-2}	6.48×10^{-2}	2.79×10^{-1}	1.14×10^0	5.02×10^0	2.31×10^1
awknn	5.51×10^{-5}	2.47×10^{-4}	5.39×10^{-4}	2.09×10^{-3}	7.55×10^{-3}	2.96×10^{-2}	4.07×10^{-2}

Note: unit—seconds. Page: 13



Experiment: Classification Accuracy

Table 4. Average accuracy of different k values in Type-1.

Classification \ Size	2	4	8	16	32	64	128
knn	47.79	64.75	78.54	86.07	90.57	93.70	95.00
wknn	59.31 (11.52)	72.91 (8.16)	82.98 (4.44)	88.82 (2.75)	92.27 (1.70)	94.82 (1.12)	95.85 (0.85)
awknn	59.30 (11.51)	72.86 (8.11)	82.94 (4.40)	88.79 (2.72)	92.27 (1.70)	94.83 (1.13)	95.85 (0.85)

Note: unit—%.

Table 5. Average accuracy of different k values in Type-2.

Classification \ Size	2	4	8	16	32	64	128
knn	51.88	62.53	72.62	79.30	83.64	87.56	91.20
wknn	59.21 (7.33)	68.11 (5.58)	76.17 (3.55)	81.85 (2.55)	85.98 (2.34)	89.77 (2.21)	92.84 (1.64)
awknn	59.20 (7.32)	68.11 (5.58)	76.17 (3.55)	81.84 (2.54)	85.98 (2.34)	89.77 (2.21)	92.77 (1.57)

Note: unit—%.