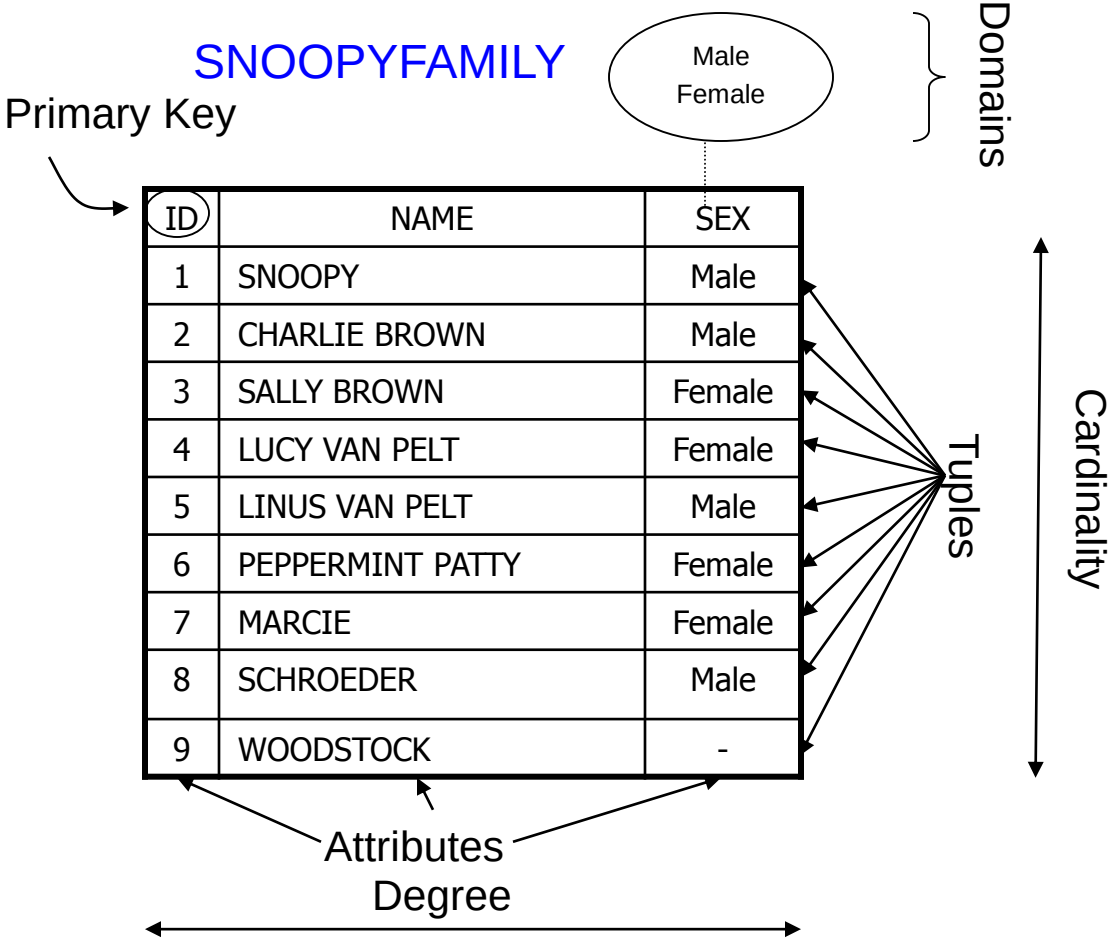
資料庫系統實驗室

指導教授：張玉盈

Relational Database



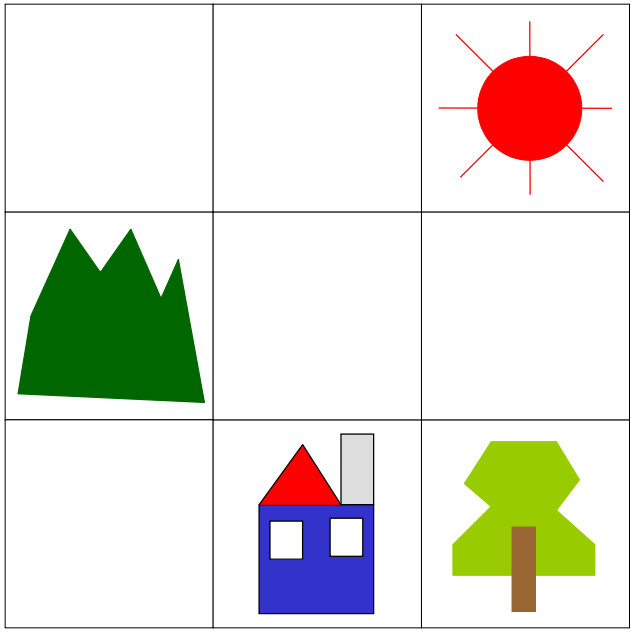
- 利用SQL做查詢：

```
Select NAME
From SNOOPYFAMILY
Where SEX = 'Male';
```

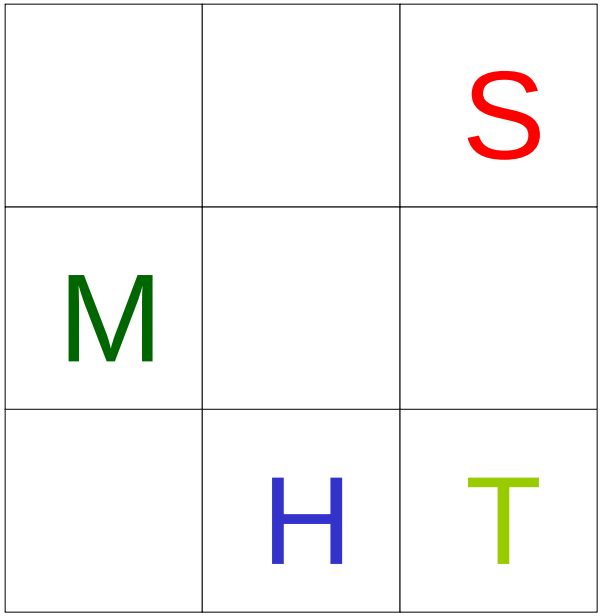
- 結果：

ID	NAME	SEX
1	SNOOPY	Male
2	CHARLIE BROWN	Male
5	LINUS VAN PELT	Male
8	SCHROEDER	Male

Image Databases



(a) An image picture



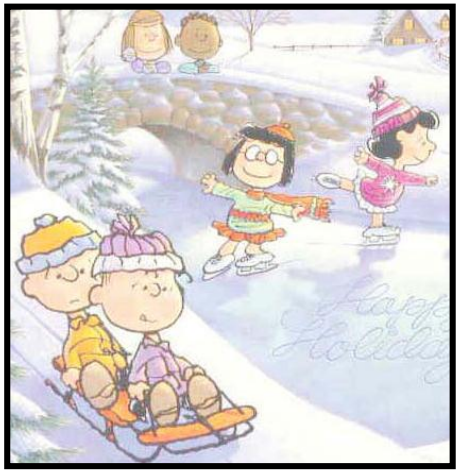
(b) The corresponding symblic representation

2D String : x : M<H<T=S

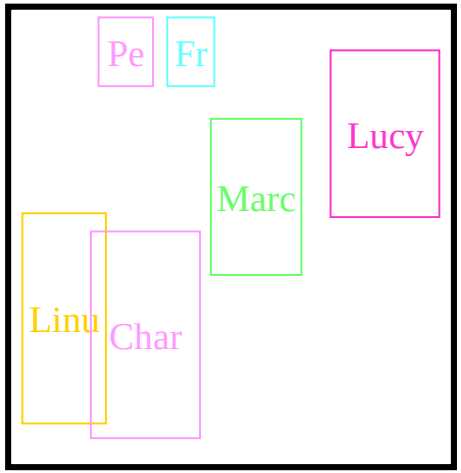
y : H=T<M<S

Image Database

- 應用層面：辦公室自動化、電腦輔助設計、醫學影像擷取…等等。
- 影像資料庫中的查詢(Queries)：
  - Spatial Reasoning(空間推理)：在一張影像中推論兩兩物件之間的空間關係。
  - Pictorial Query(圖像查詢)：允許使用者給予特定的空間關係以查詢相對應的影像。
  - Similarity Retrieval(圖形相似擷取)：藉由使用者所提供的資訊在影像資料庫中找尋出最相似的圖形。

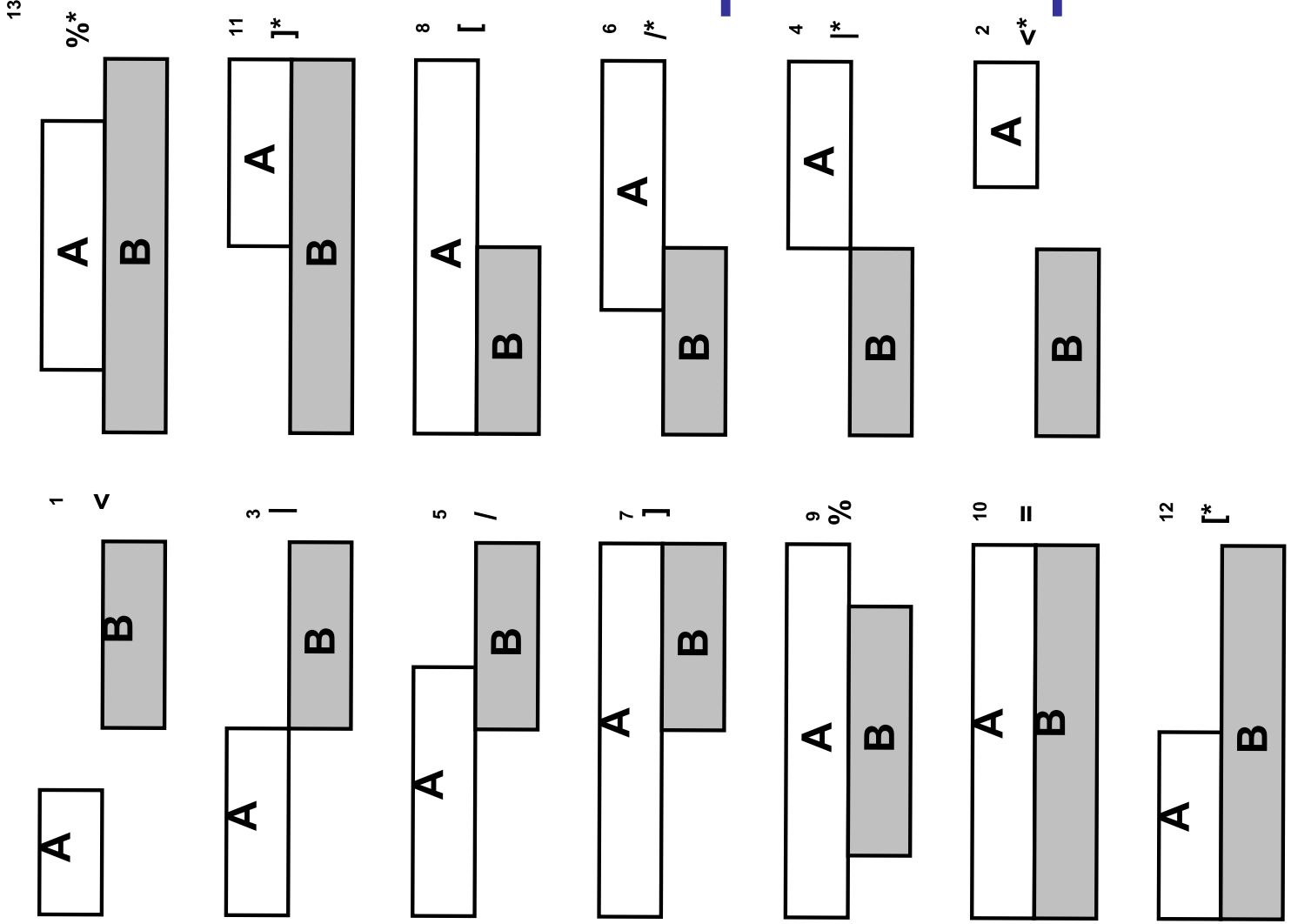


(a) An image picture

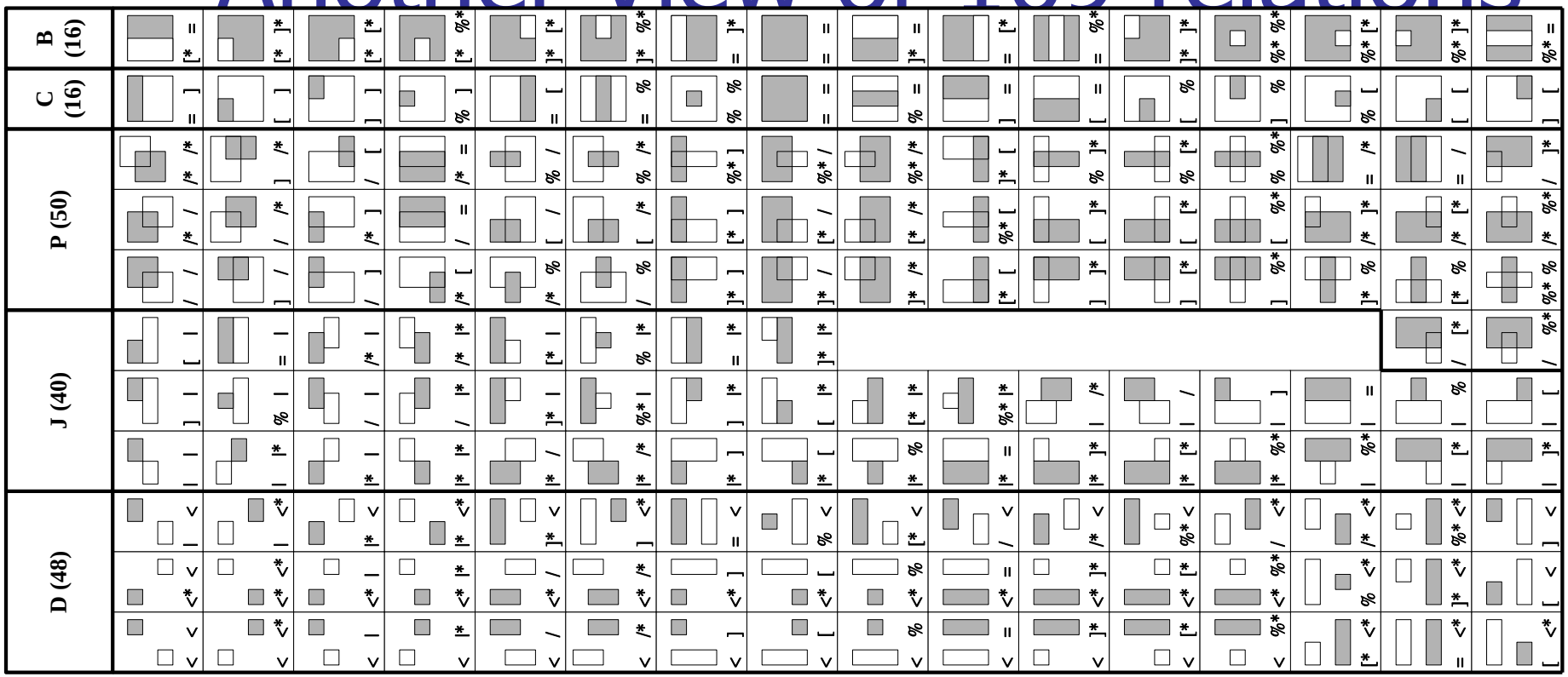


(b) Symbolic Picture

Uids of 13 spatial operators

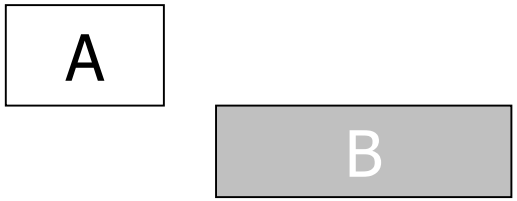


Another View of 169 relations

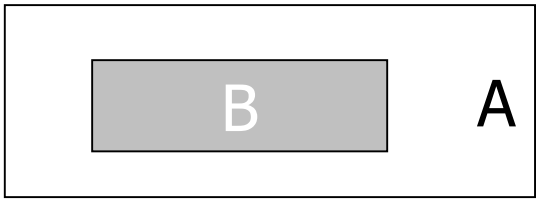


➢5 Category Relationships(C_{AB})

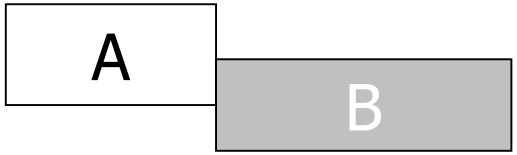
Disjoin :



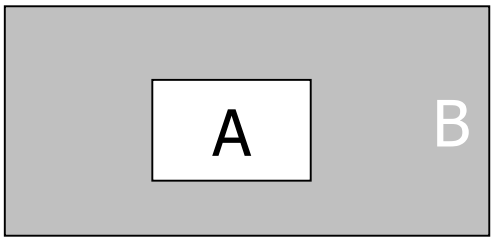
Contain :



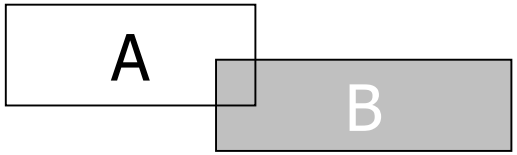
Meet :



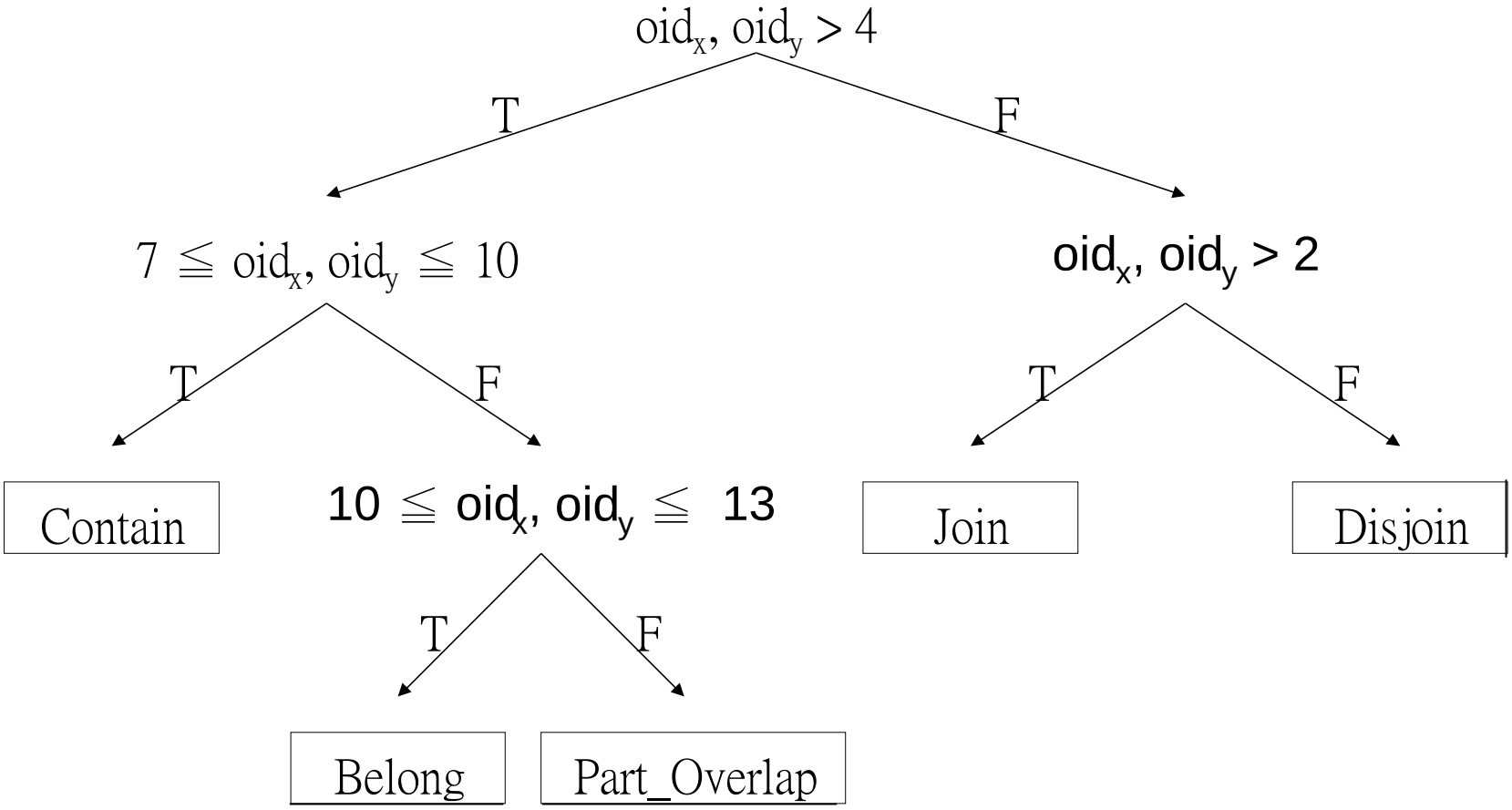
Inside :



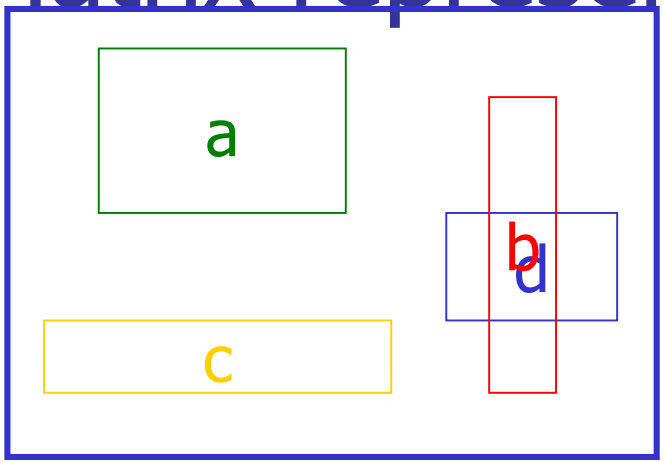
Partly Overlap :



➢Decision tree of the CATEGORY function



➢UID Matrix representation(cont.)



f_1

	a	b	c	d
a	0	/*	<*	/*
b	<	0	/*	%
c	%*	<*	0	<
d	<	%*	<	0

	a	b	c	d
a	0	6	2	6
b	1	0	6	9
c	13	2	0	1
d	1	13	1	0

➢Similarity Retrieval based on the UID Matrix(1)

- **Definition1** Picture *f'* is a type-*i* unit picture of *f*, if

  (1) *f'* is a picture containing the two objects A and B, represented as *x*: A $r^{x'}_{A,B}$ B, *y*: A $r^{y'}_{A,B}$ B.

  (2) A and B are also contained in *f*.

  (3) the relationships between A and B in *f* are represented as *x*: A $r^{x}_{A,B}$ B, and *y*: A $r^{y}_{A,B}$ B.

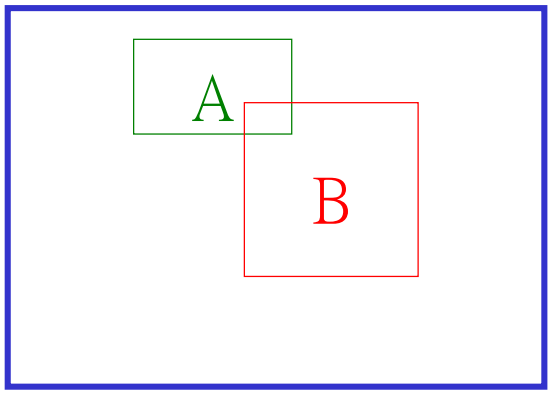
  Then,

  (Type-0): Category($r^{x'}_{A,B}$, $r^{y'}_{A,B}$)

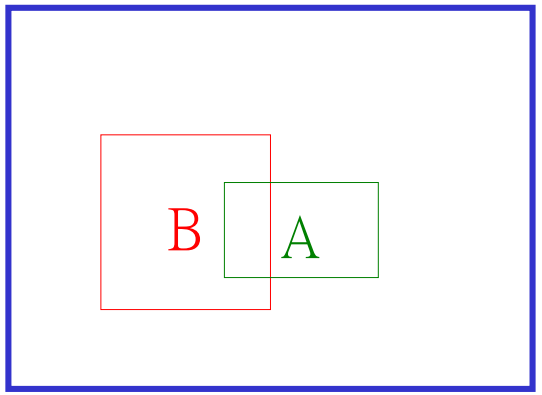
  (Type-1): (Type-0) and ($r^{x'}_{A,B} = r^{x}_{A,B}$ or $r^{y'}_{A,B} = r^{y}_{A,B}$)

  (Type-2): $r^{x'}_{A,B} = r^{x}_{A,B}$ and $r^{y'}_{A,B} = r^{y}_{A,B}$

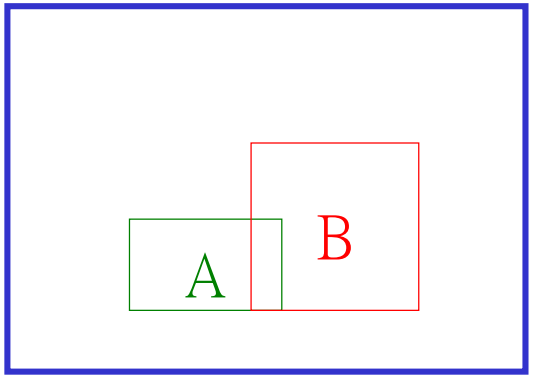
➢3 type-i similarities



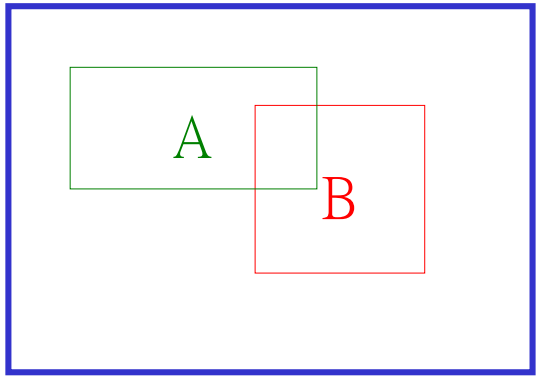
f(A/B, A/*B)



type-0(A/*B, A%*B)



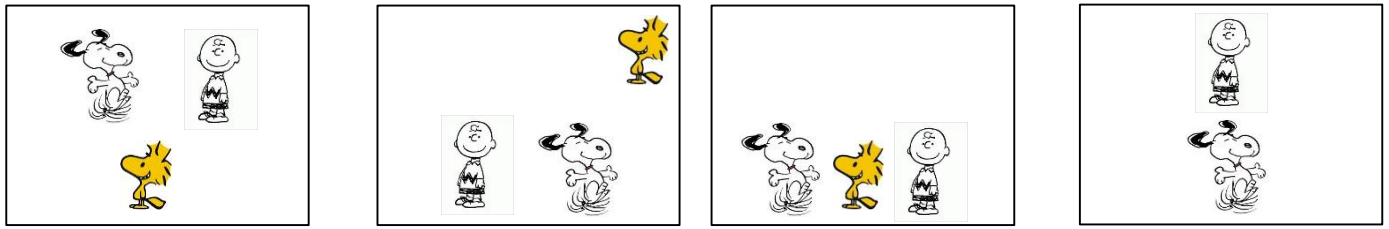
type-1 (A/B, A[*B)



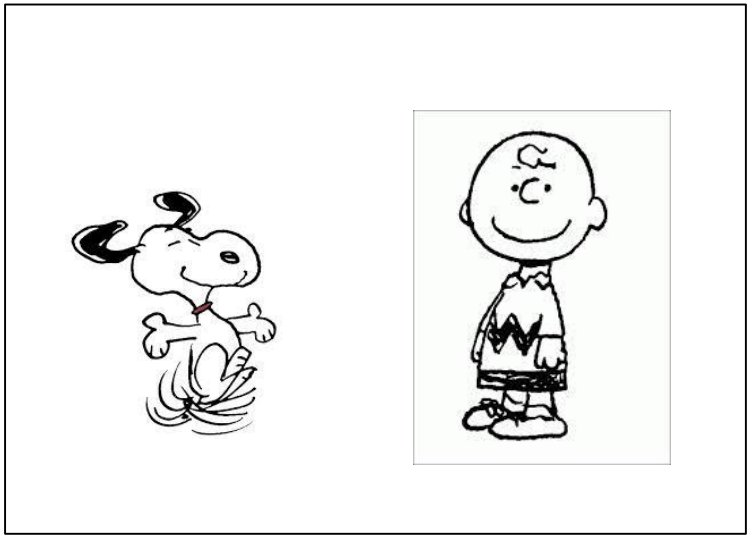
type-2 (A/B, A/*B)

Image Mining:

Finding Frequent Patterns in Image Databases



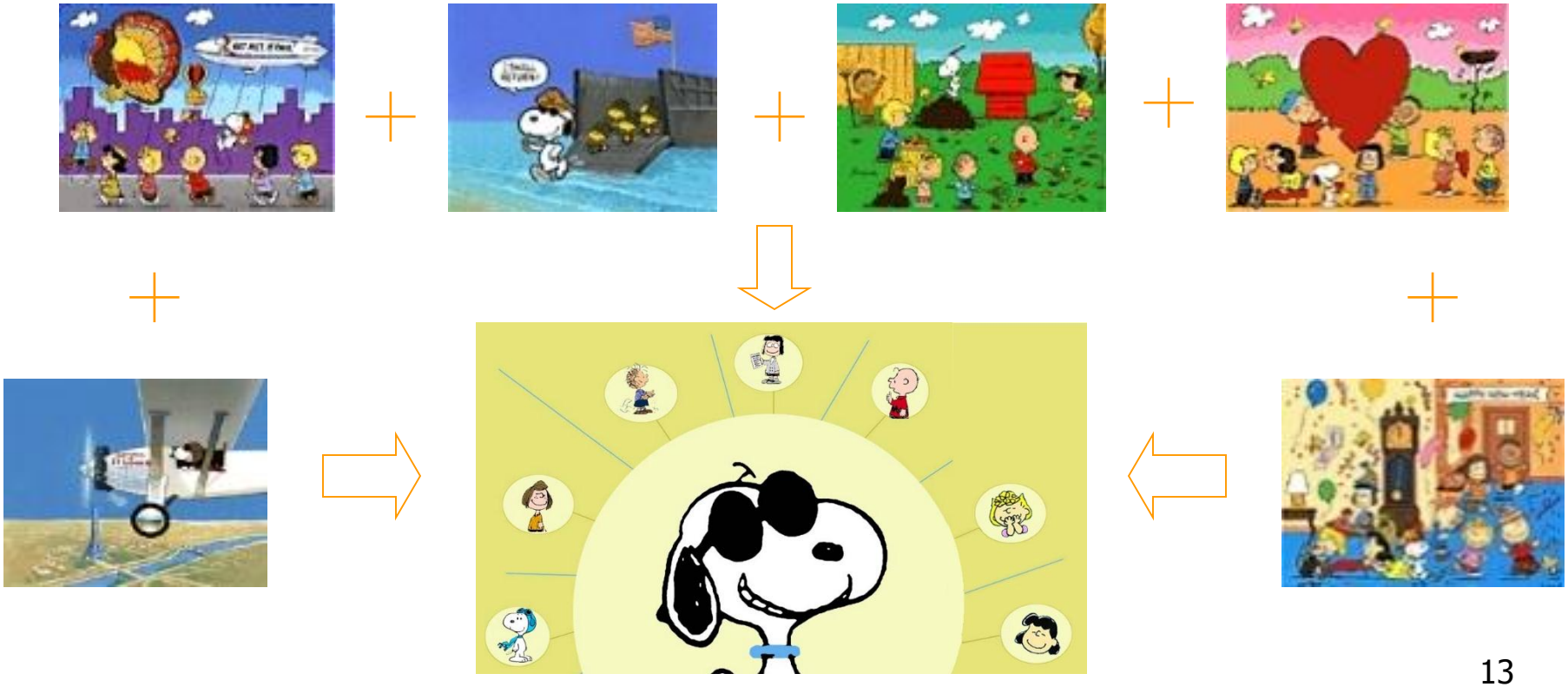
Setting the *minimum support* to ½ .



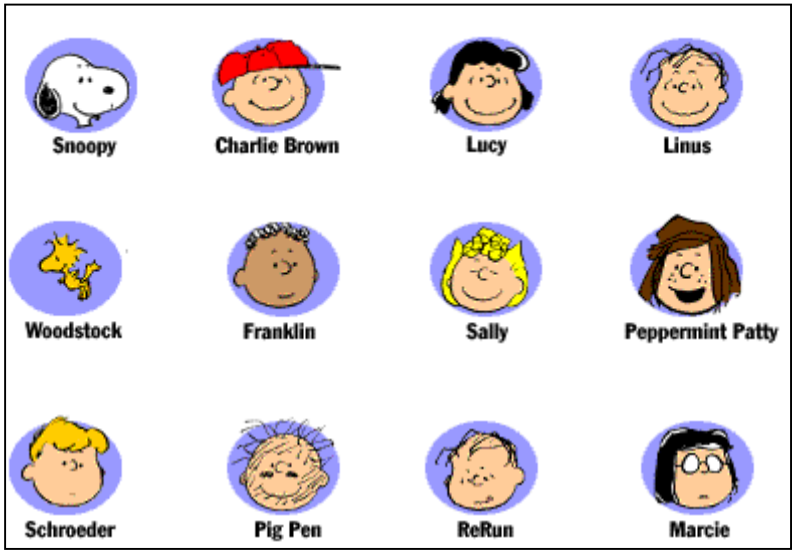
Charlie Brown often appears to the right of *Snoopy*.

Video : Image + Time

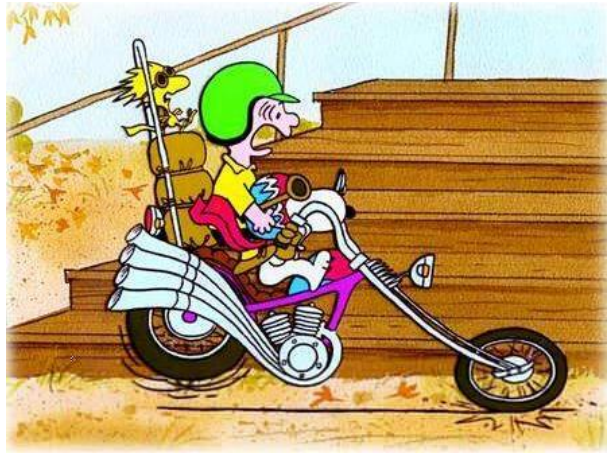
範例： 一幕幕的Snoopy影像，編織成一部精彩的Snoopy影片



Multimedia Database



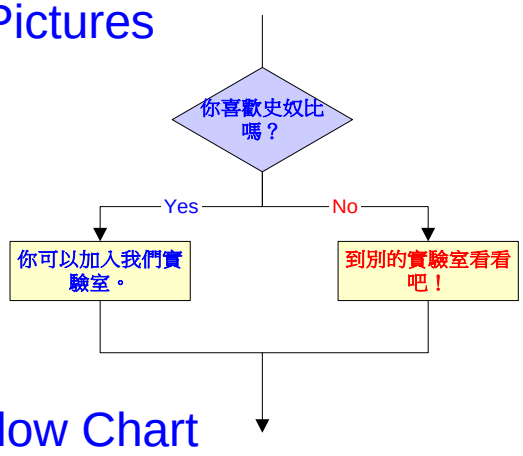
– Pictures with the depicted texts



– Pictures

– Voice

– Video



– Flow Chart

Spatial Database :
Nearest Neighbor Query

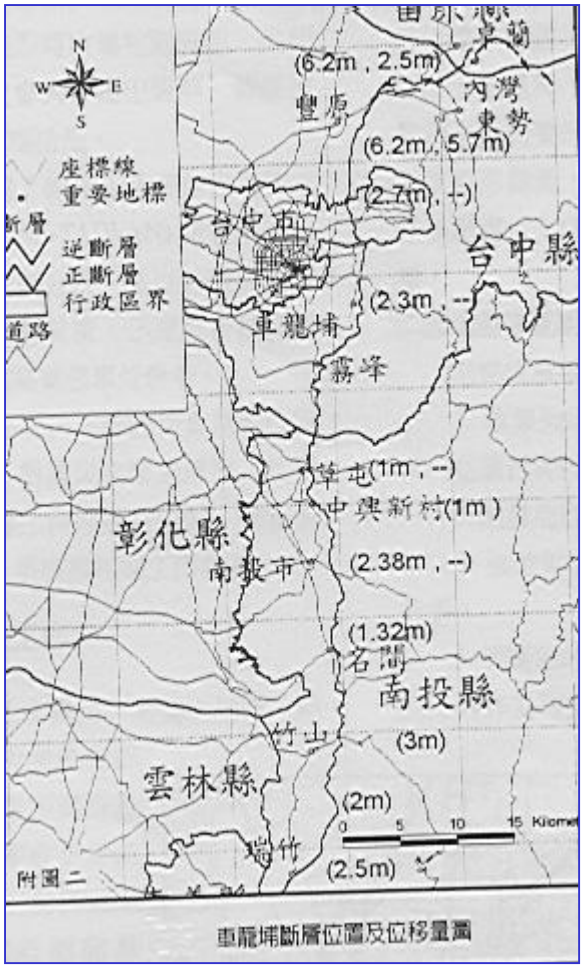
Query Types

1. 精確比對查詢：
   哪一個城市位在北緯43度與西經88度？

2. 部分比對查詢：
   哪些城市的緯度屬於北緯39度43分？

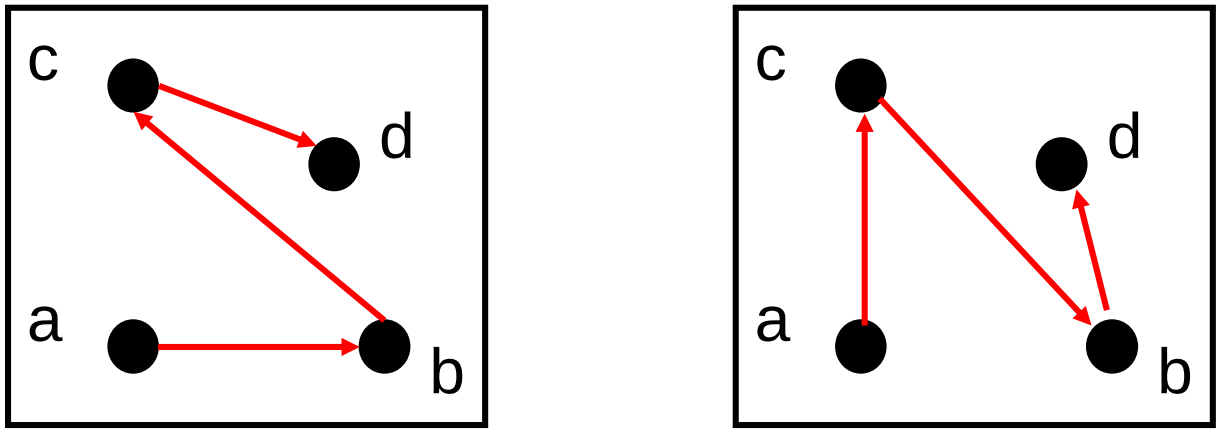
3. 給定範圍查詢：
   哪些城市的經緯度介於北緯39度43分至43度與西經53度至58度之間？

4. 近似比對查詢：
   最靠近東勢鎮的城市是？



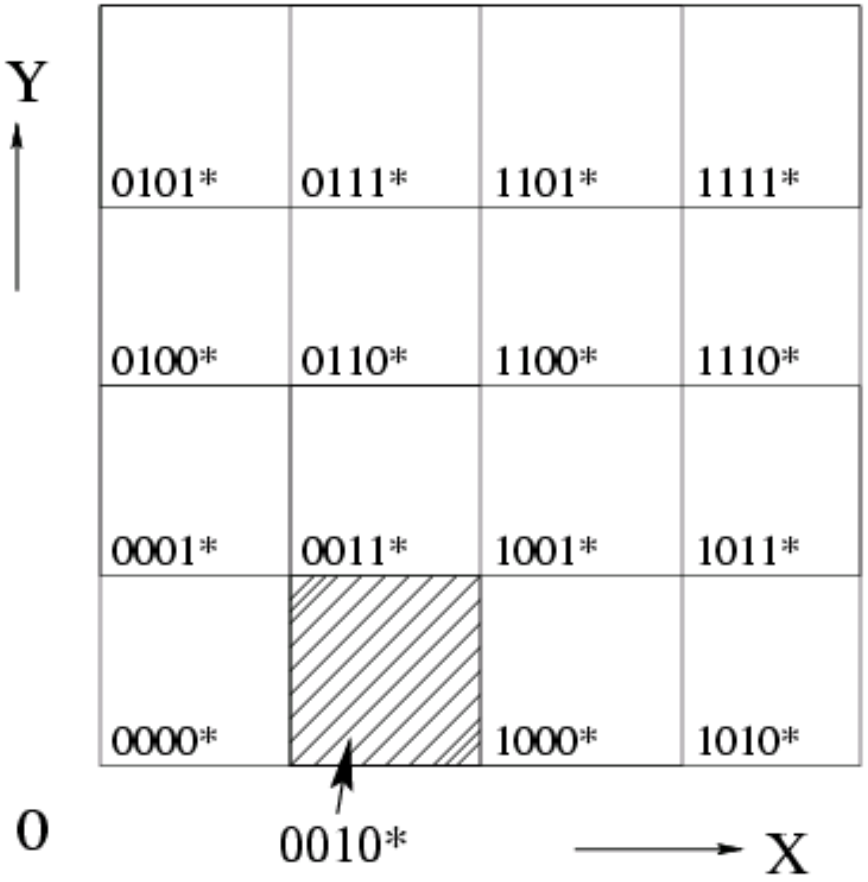
Difficulty

- No total ordering of spatial data objects that preserves the spatial proximity.



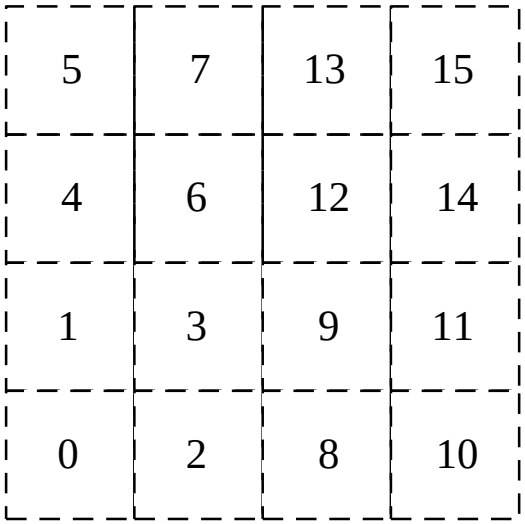
a b c d ? / a c b d ?

Space Decomposition and DZ expression

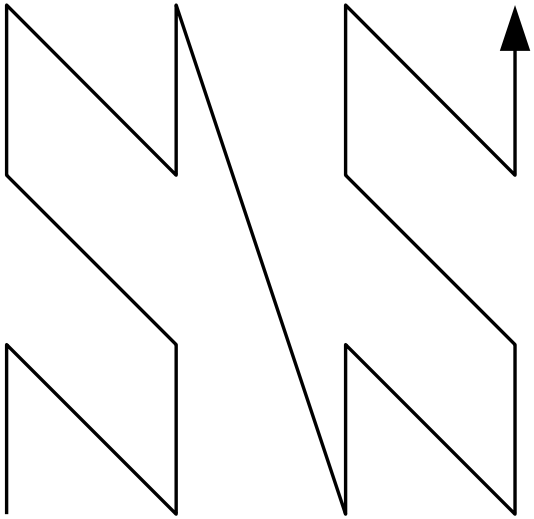


0 - 0000*
1 - 0001*
2 - 0010*
3 - 0011*
4 - 0100*
5 - 0101*
6 - 0110*
7 - 0111*
8 - 1000*
9 - 1001*
10 - 1010*
11 - 1011*
12 - 1100*
13 - 1101*
14 - 1110*
15 - 1111*

The Bucket-Numbering Scheme

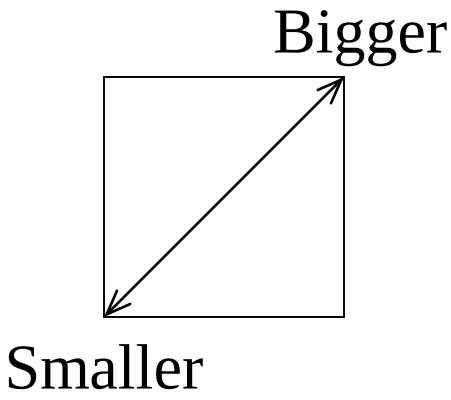


(a)



(b)

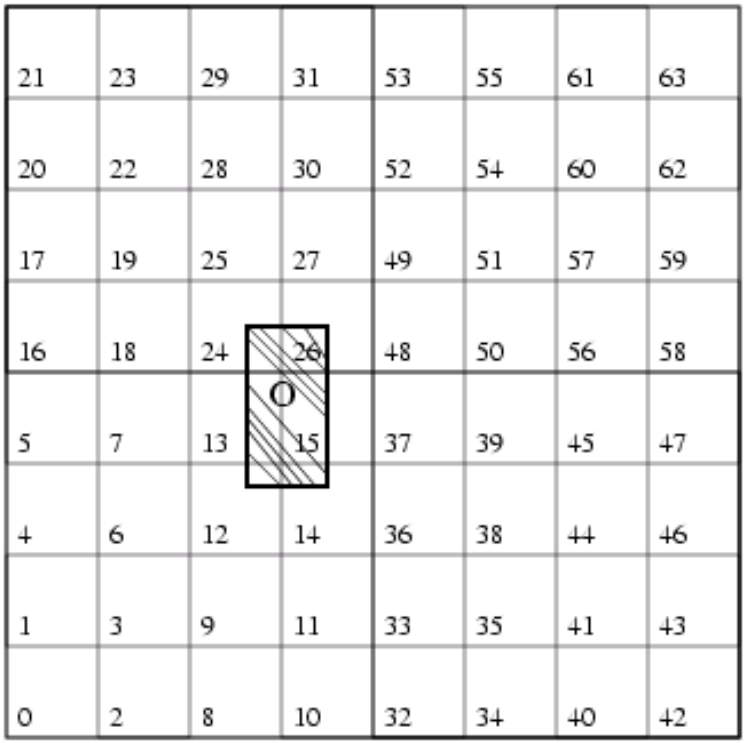
N-order Peano Curve



(c)

the uptrend of the bucket numbers of an object

Example



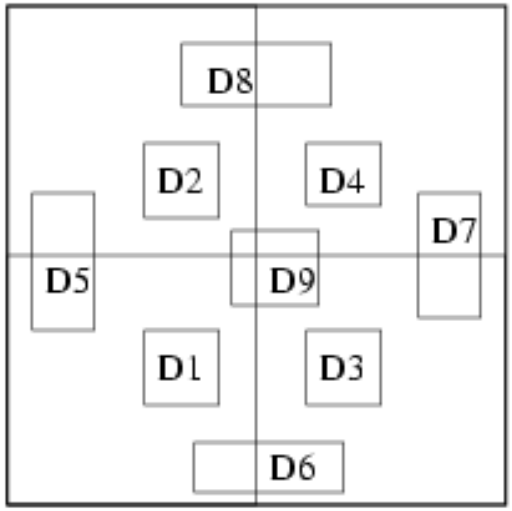
O(l,u) = (12,26)

The total number of buckets depends on the expected number of data objects.

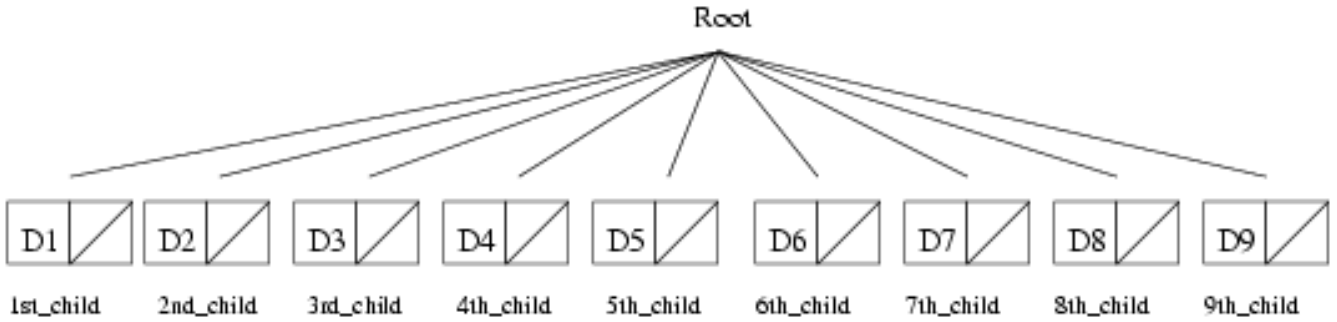
maximum bucket number:

Max_bucket = 63

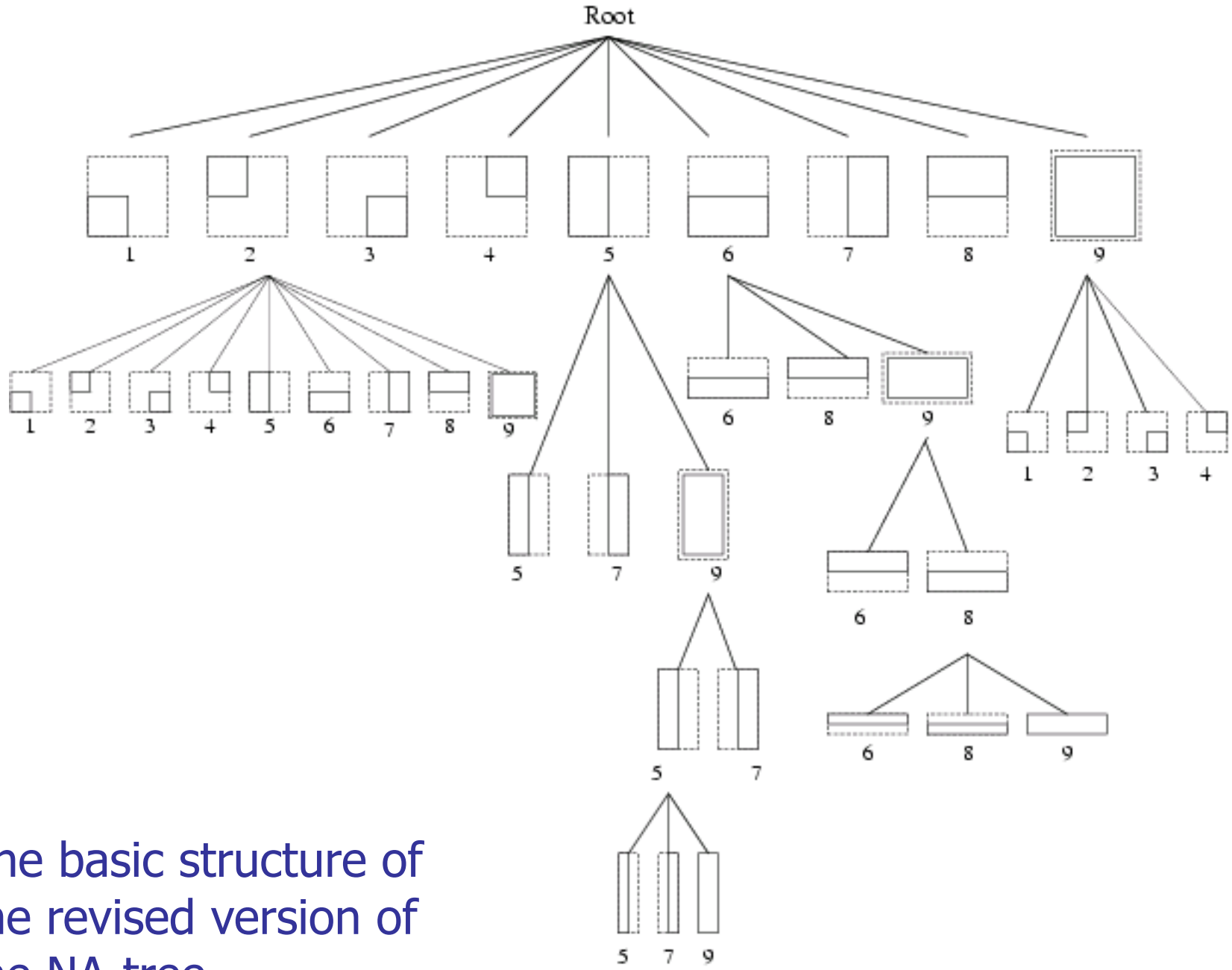
Example



(a) the data



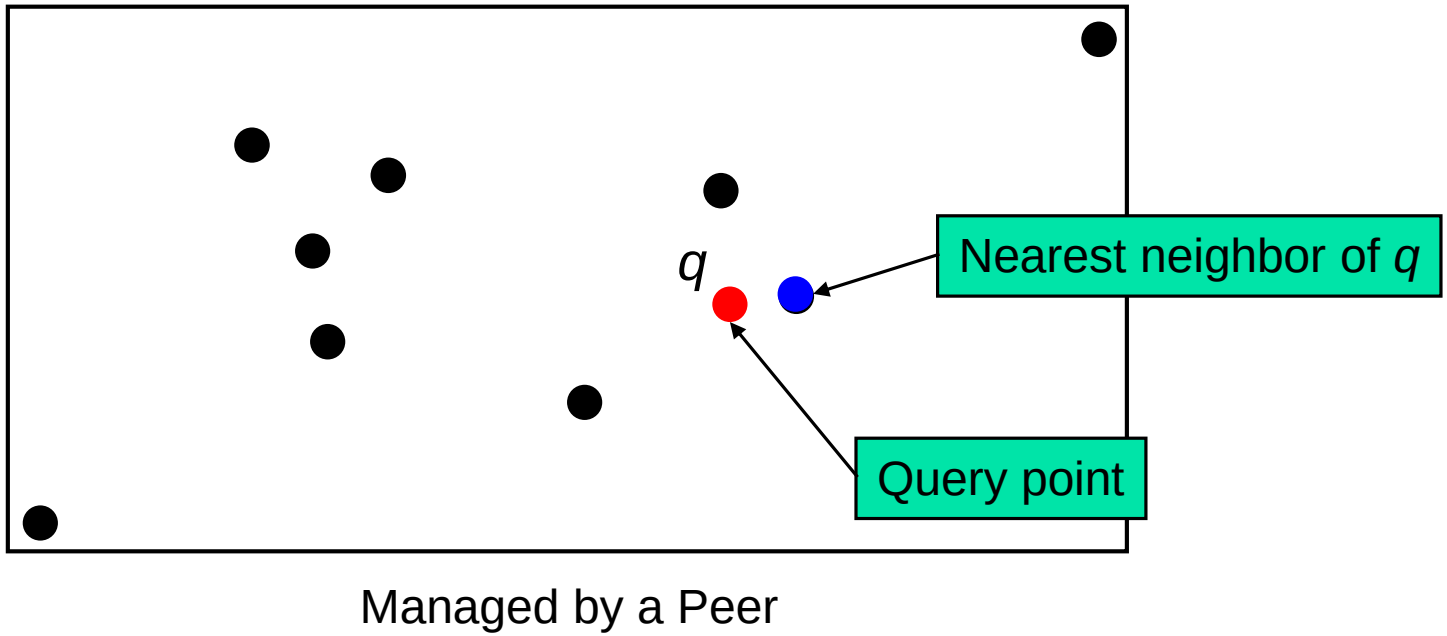
(b) the corresponding NA-tree structure (bucket_capacity = 2)



The basic structure of the revised version of the NA-tree

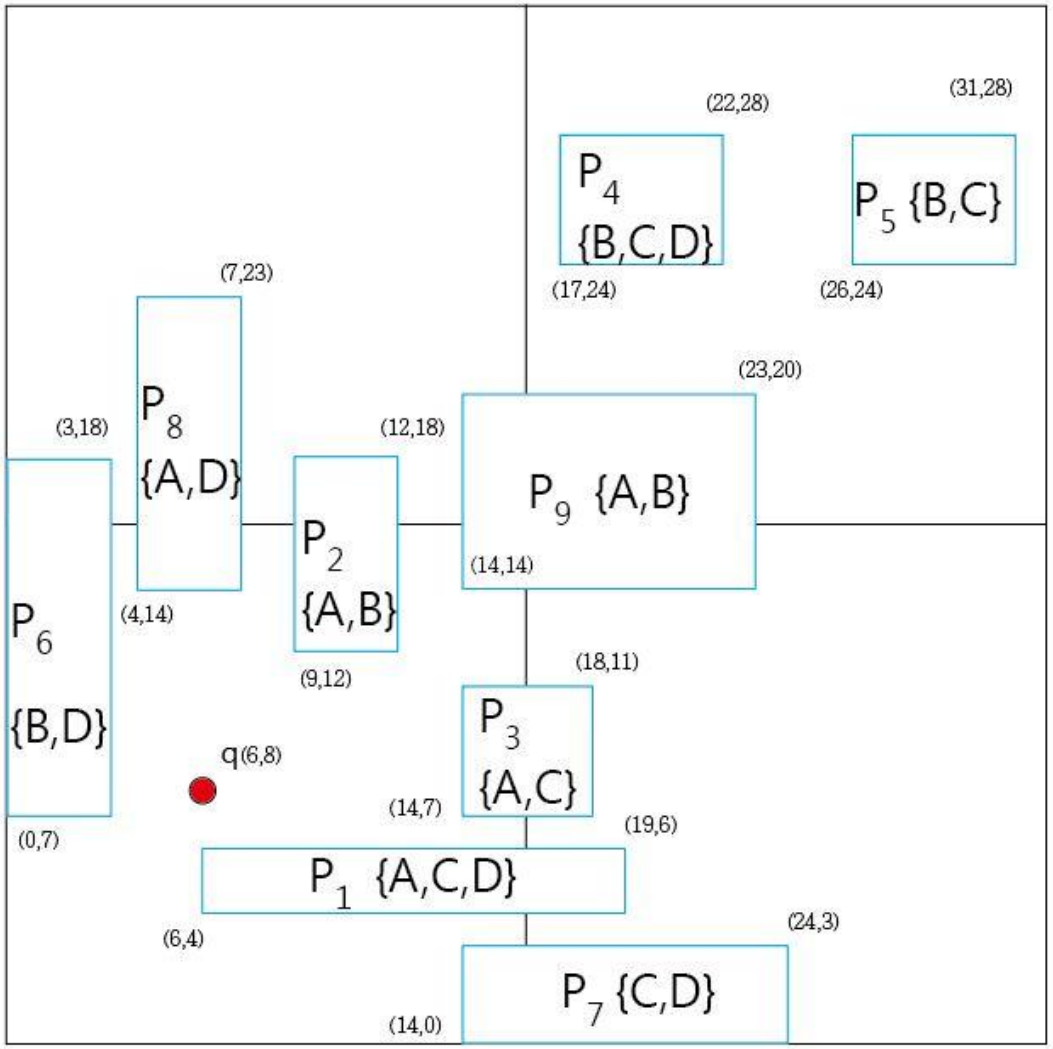
NN (Nearest Neighbor)

- NN problem is to find the nearest neighbor of *q* (query point).



Spatial Databases:

KNN Keyword Query



Where are the 2 nearest points with keywords B and C?

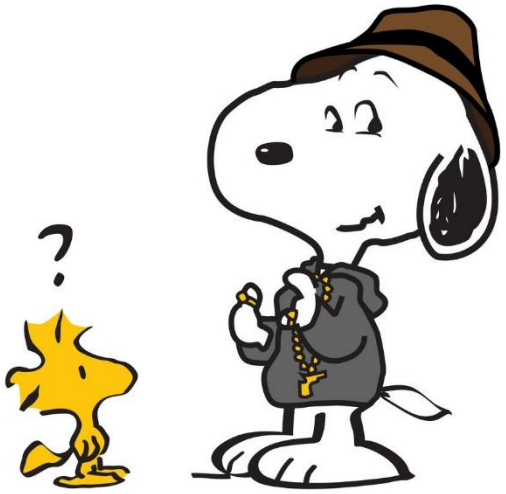
Road Network Databases:

K Nearest Neighbor Query

Where are the 3 nearest restaurants?

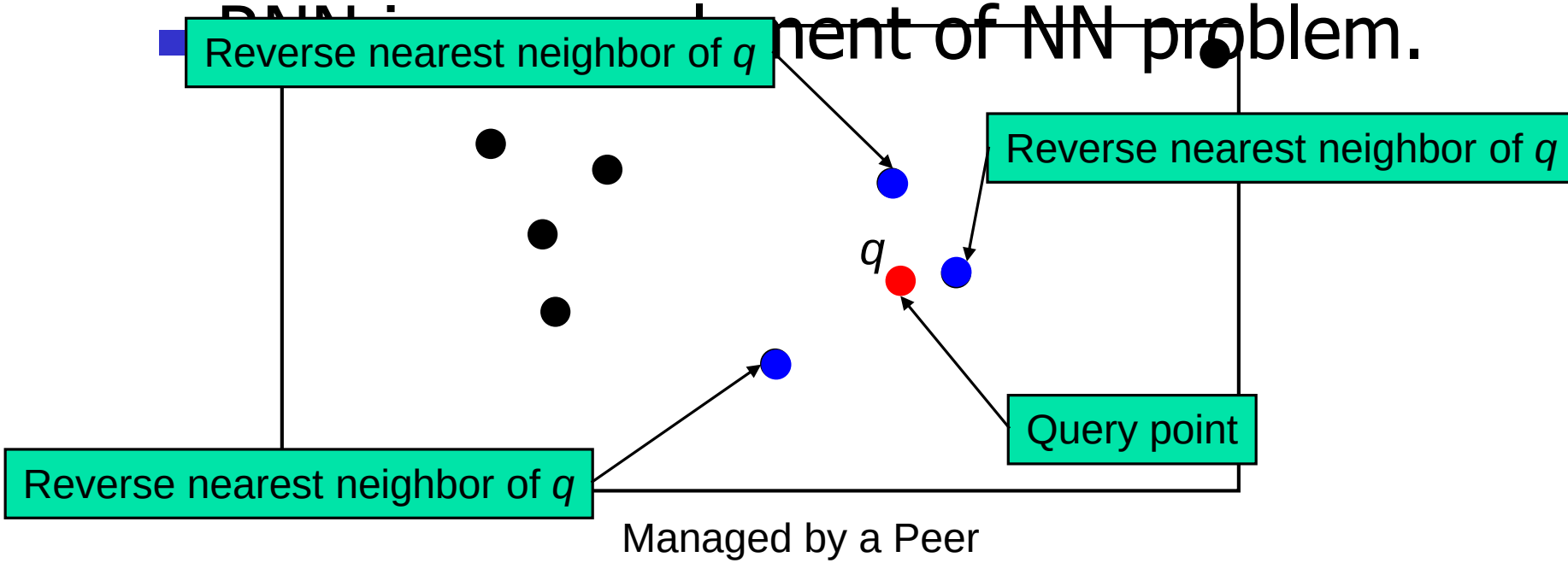
Spatial Databases:
Top-*k* Spatial Keyword Query

Where are the top-1 'Snoopy hotel' near Kaohsiung?

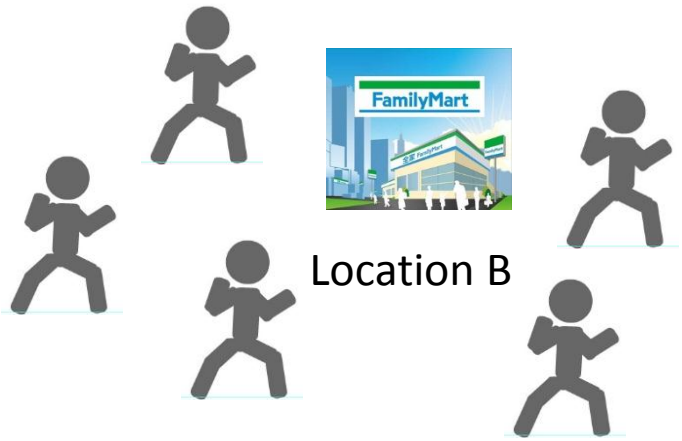


RNN (Reversed NN)

- The *q* is the nearest neighbor of the blue points.
- RNN is a complement of NN problem.



- Reverse Nearest Neighbor(RNN) Query means : to obtain the objects which treat the query as their nearest neighbor.
- Application : Business strategy



Five residents treat Location B as their NN.
Three residents treat Location A as their NN.
Location B is a better place to run the store.

Query q


Residents

- Reverse Nearest Neighbor(RNN) Query means : to obtain the objects which treat the query as their nearest neighbor.
- Application : Traffic police

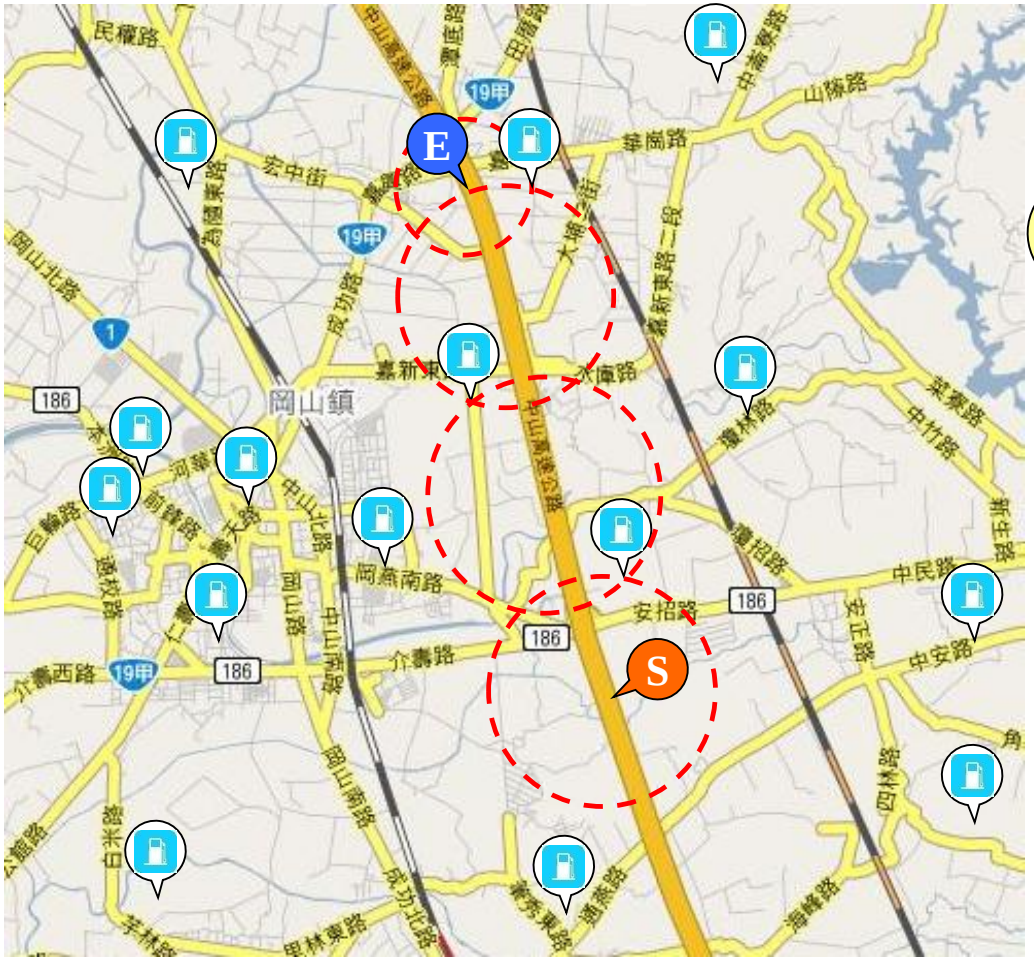
Five cars treat Location A as their NN.
Three cars treat Location B as their NN.
Location A is a better place to the police for patrol.


Query q

A
Query move


Cars

Spatial Database :
Continuous Nearest Neighbor Query



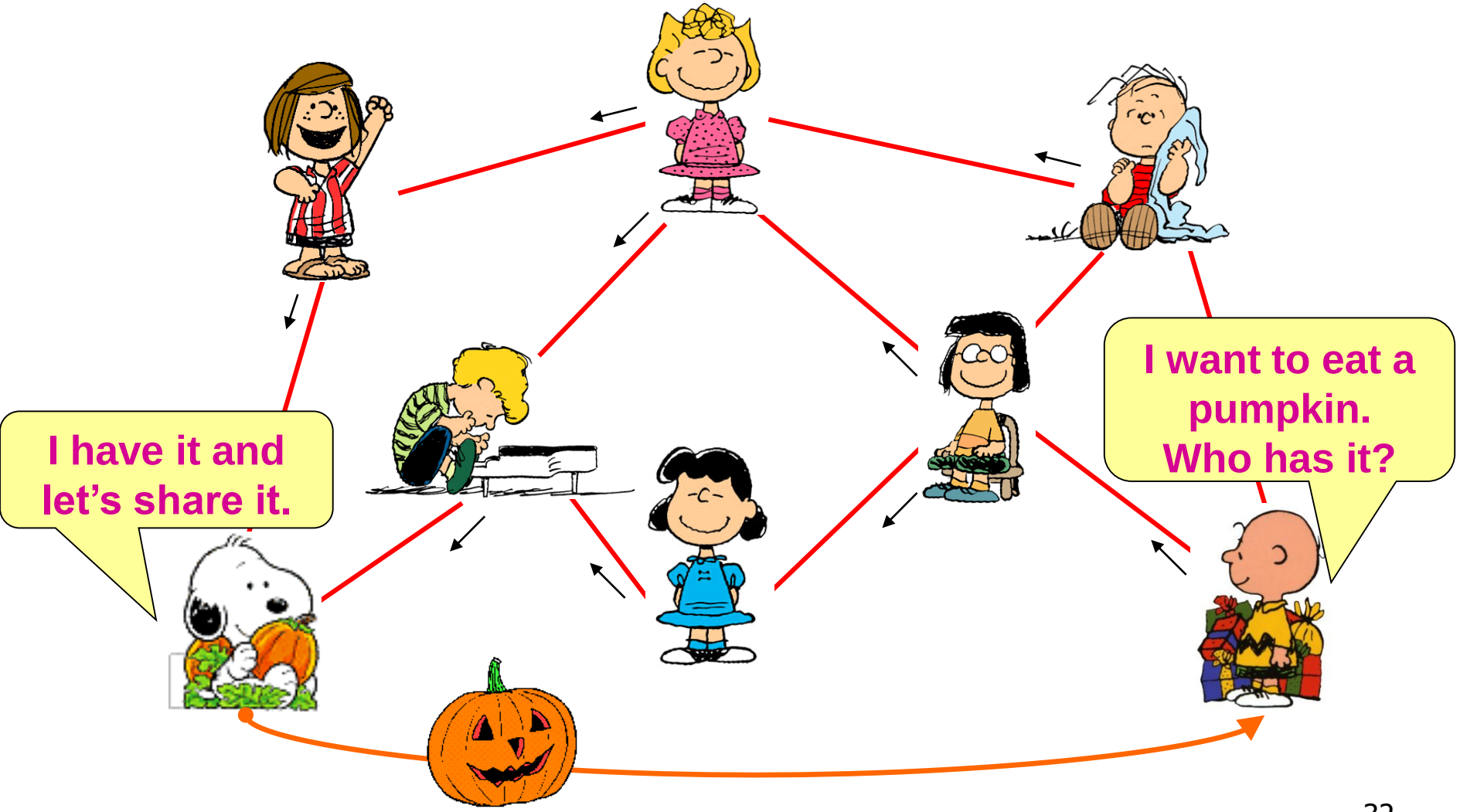
Find the nearest gas stations from the starting point to the ending point.

Spatio-temporal Dat

Where is the available gas station around my location after 20 minutes?

What is the traffic condition ahead of me during the next 30 minutes?

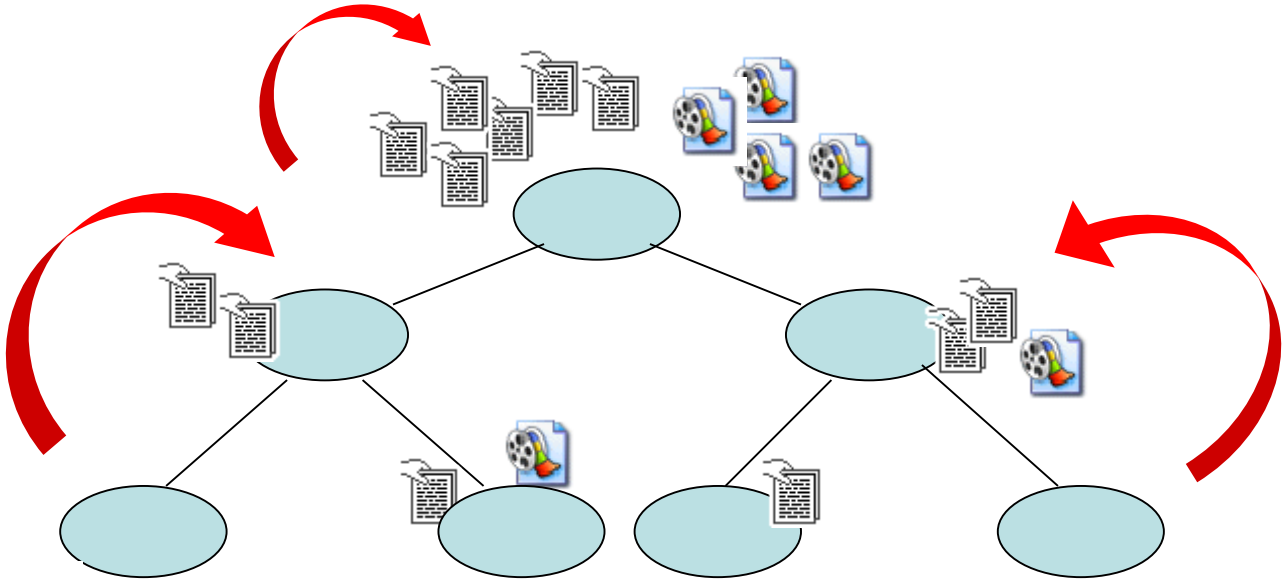
P2P System



Client-server *vs.* Peer-to-Peer network

- Example : How to find an object in the network
  - Client-server approach
    - Use a big server store objects and provide a directory for look up.
  - Peer-to-Peer approach
    - Data are fully distributed.
    - Each peer acts as both a client and a server.
    - By asking.

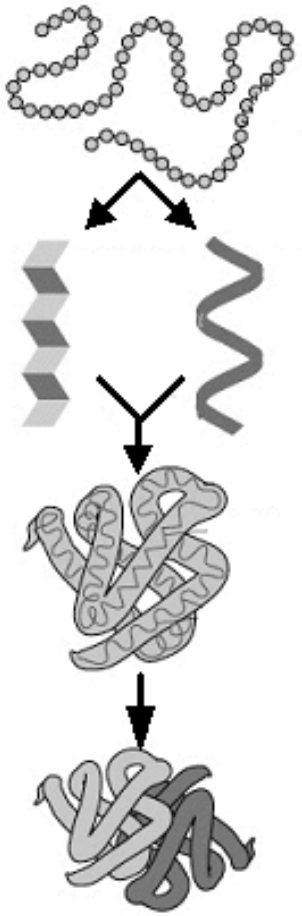
Data Grids



I want *File-A*.

I want *File-X*.

Protein Database

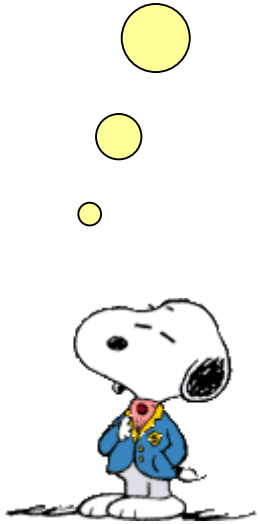


Sequence 1 KGGAKRHRKIL
Sequence 2 KVGAKRHSKRS
Sequence 3 KVGAKRHSRKS
Sequence 4 KGGAKRHRKVL

判斷蛋白質
所屬家族

判斷蛋白質
功能

Find the patterns from the protein database.

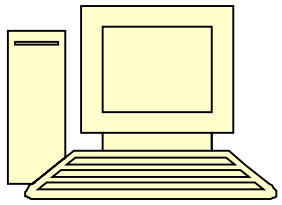


Data Mining

Peanuts Supermarket

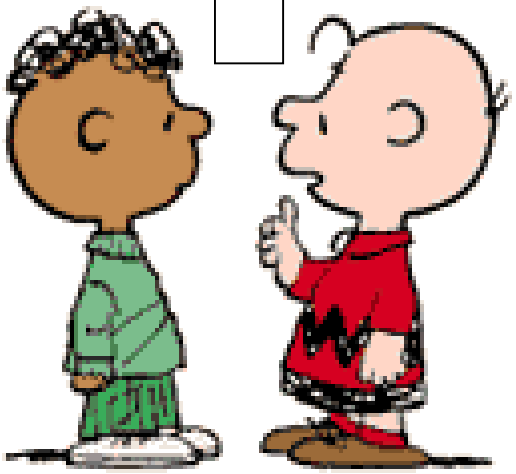
大家排隊來結帳

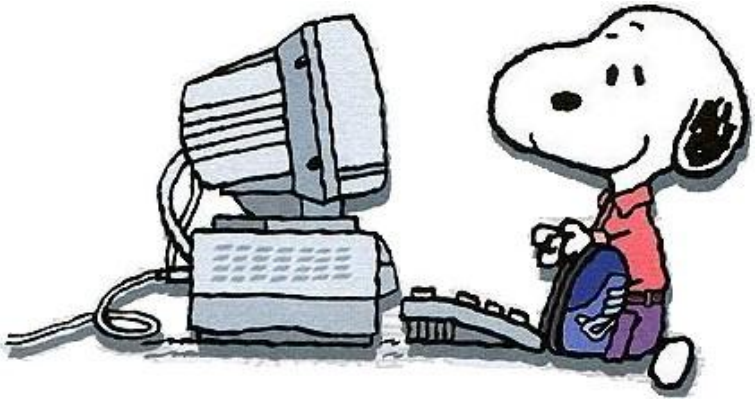
收銀台



PC

顧客通常在買麵包時也會買牛奶





利用資料挖礦的技術
對大家購買的紀錄作分析

Database *D*

TID	Items
100	A C D
200	B C E
300	A B C E
400	B E

Scan D →

C_1

Itemset	Sup.
{A}	2
{B}	3
{C}	3
{D}	1
{E}	3

L_1

Itemset	Sup.
{A}	2
{B}	3
{C}	3
{E}	3

C_2

Itemset
{A B}
{A C}
{A E}
{B C}
{B E}
{C E}

Scan D →

C_2

Itemset	Sup.
{A B}	1
{A C}	2
{A E}	1
{B C}	2
{B E}	3
{C E}	2

L_2

Itemset	Sup.
{A C}	2
{B C}	2
{B E}	3
{C E}	2

C_3

Itemset
{B C E}

Scan D →

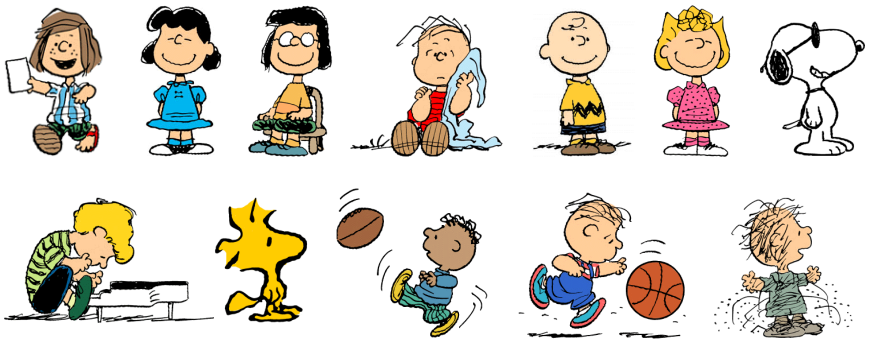
C_3

Itemset	Sup.
{B C E}	2

L_3

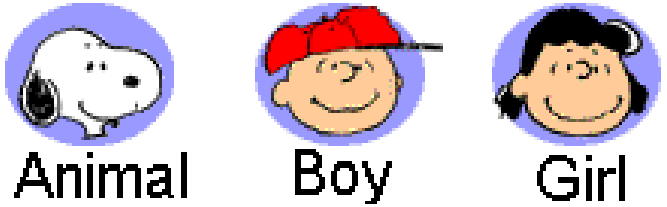
Itemset	Sup.
{B C E}	2

Data Clustering

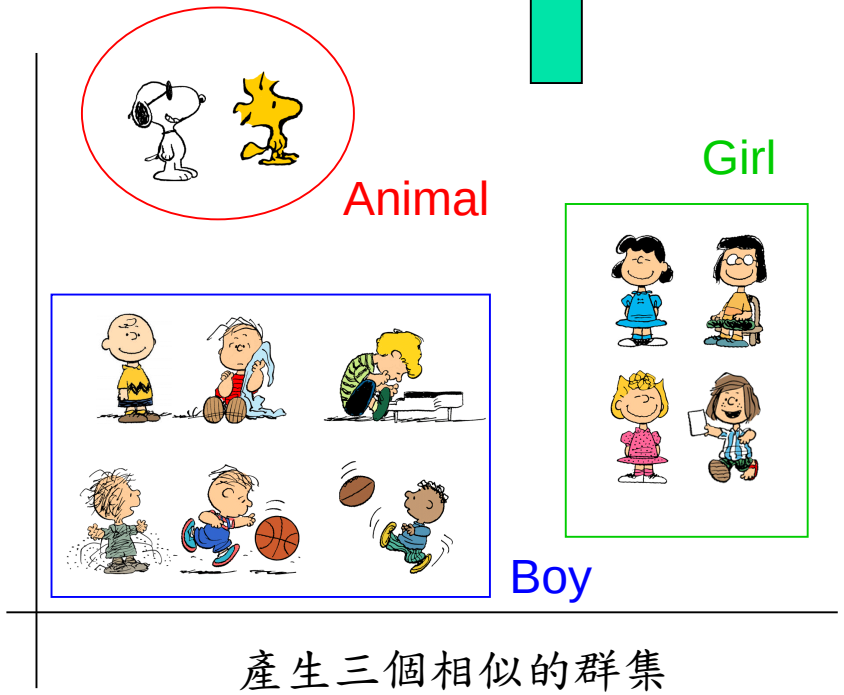


一組非常雜亂的資料，分析困難

找到資料間彼此相似的特性

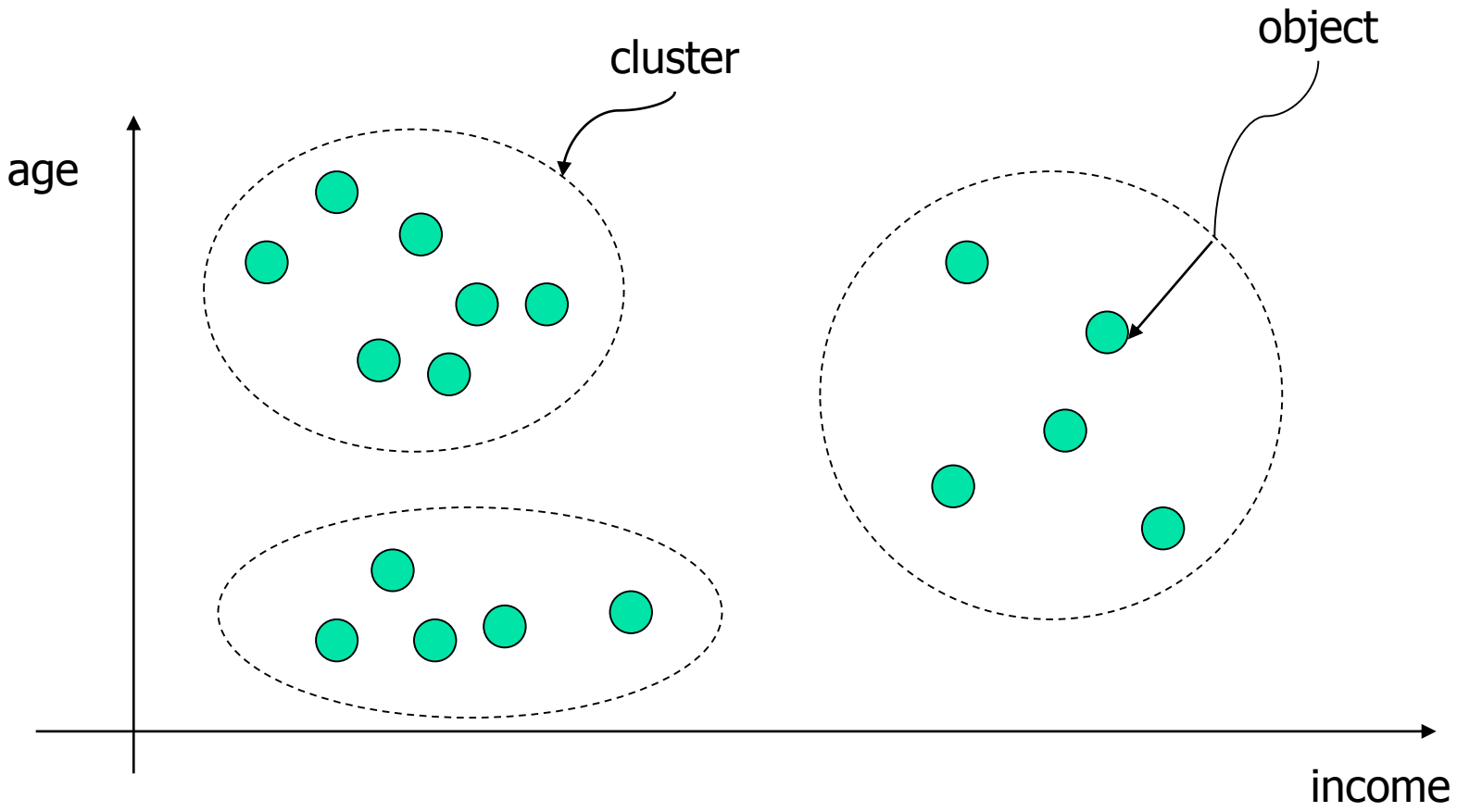


形成三個較為單純群集再做分析較為容易



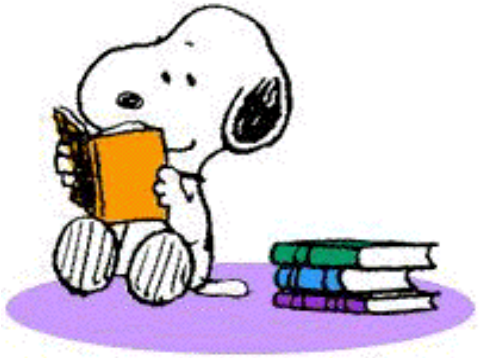
產生三個相似的群集

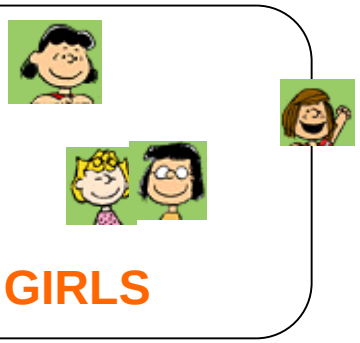
Example



Classification

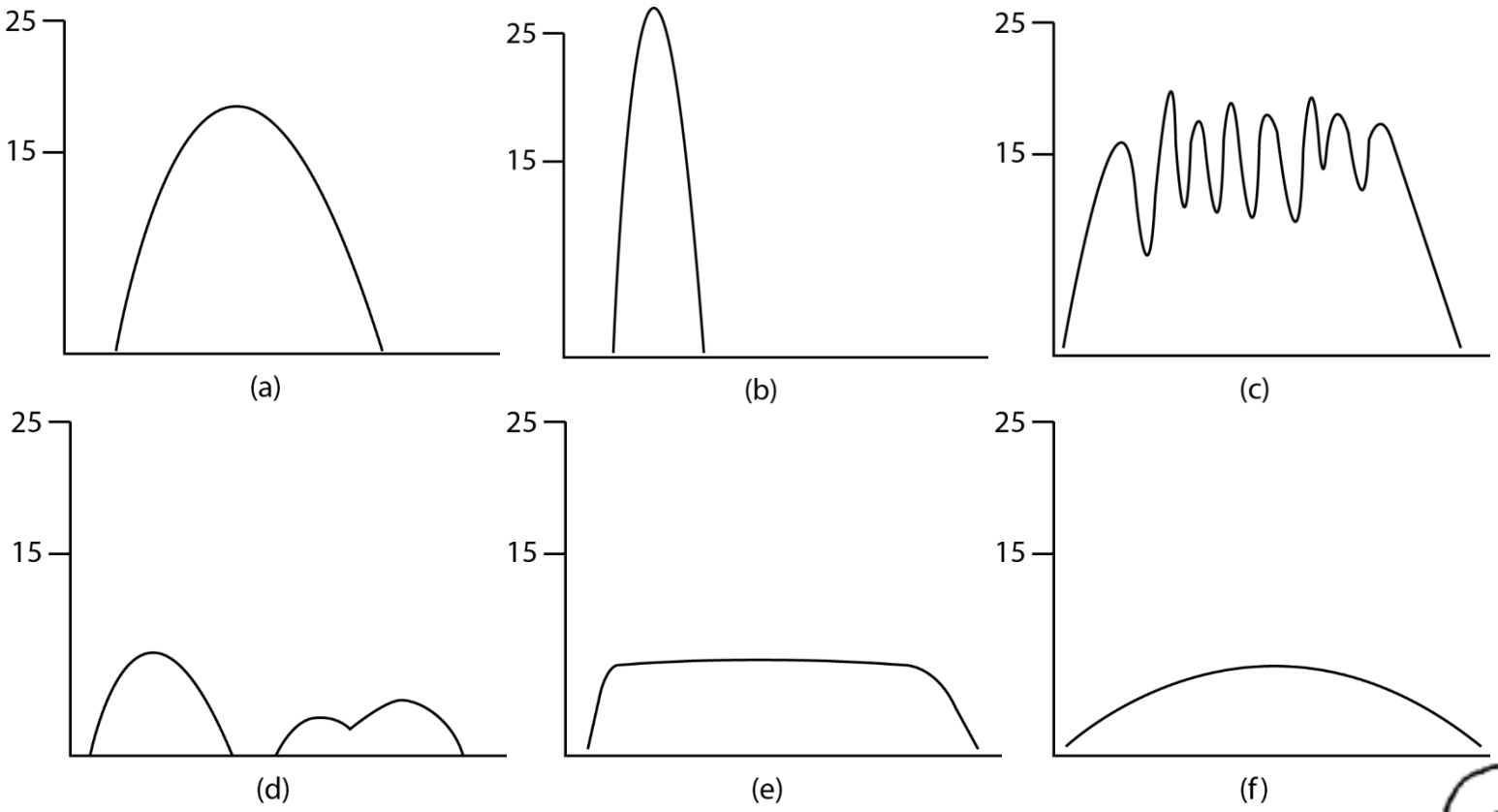
從目前的
資料中學習





對新的資料
做準確的
預測分類

Classification of Uroflowmetry Curves

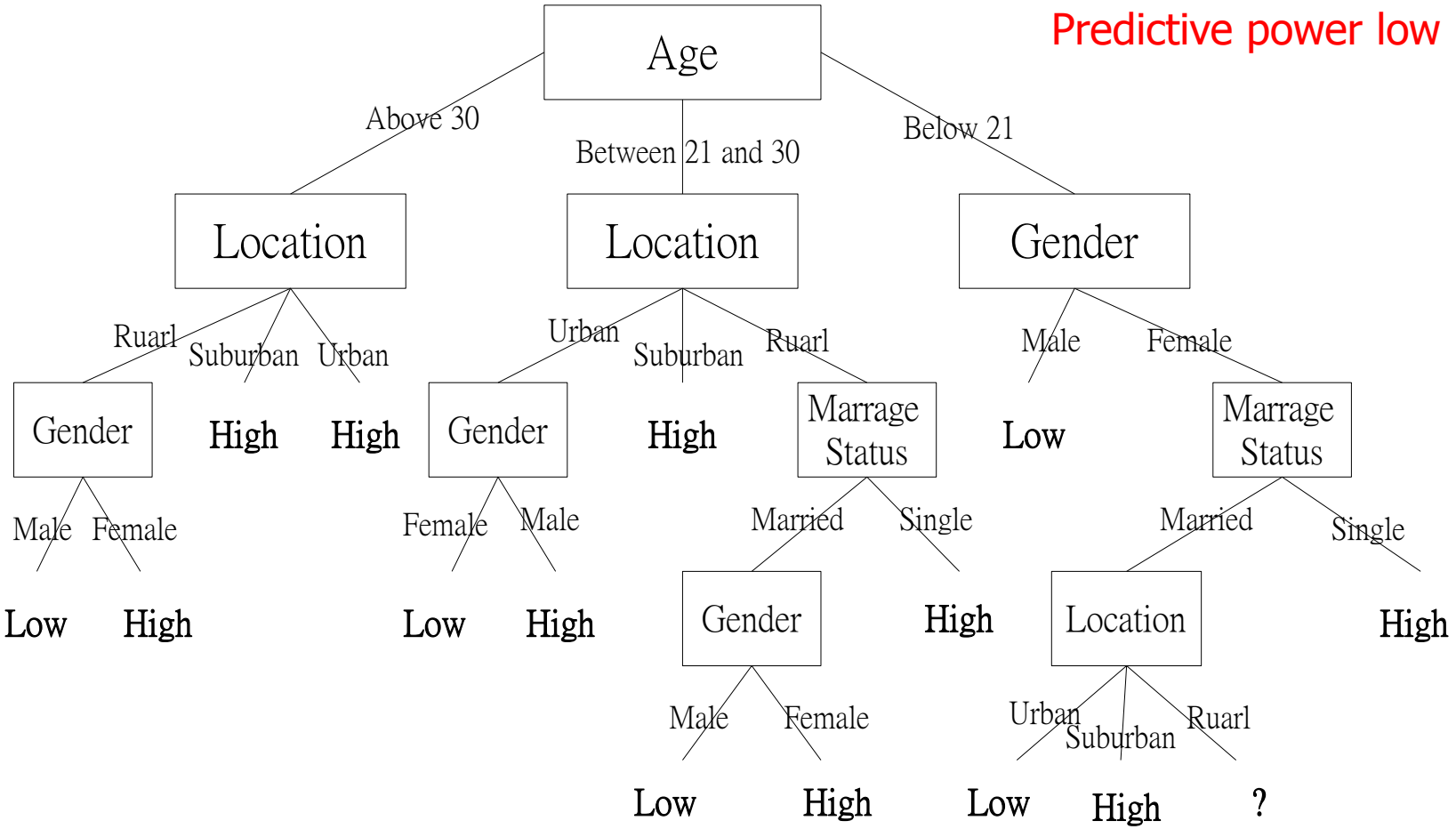


Uroflow patterns: (a) Bell-shaped; (b) Tower-shaped; (c) Staccato-shaped; (d) Interrupted-shaped; (e) Plateau-shaped; (f) Obstructive-shaped.

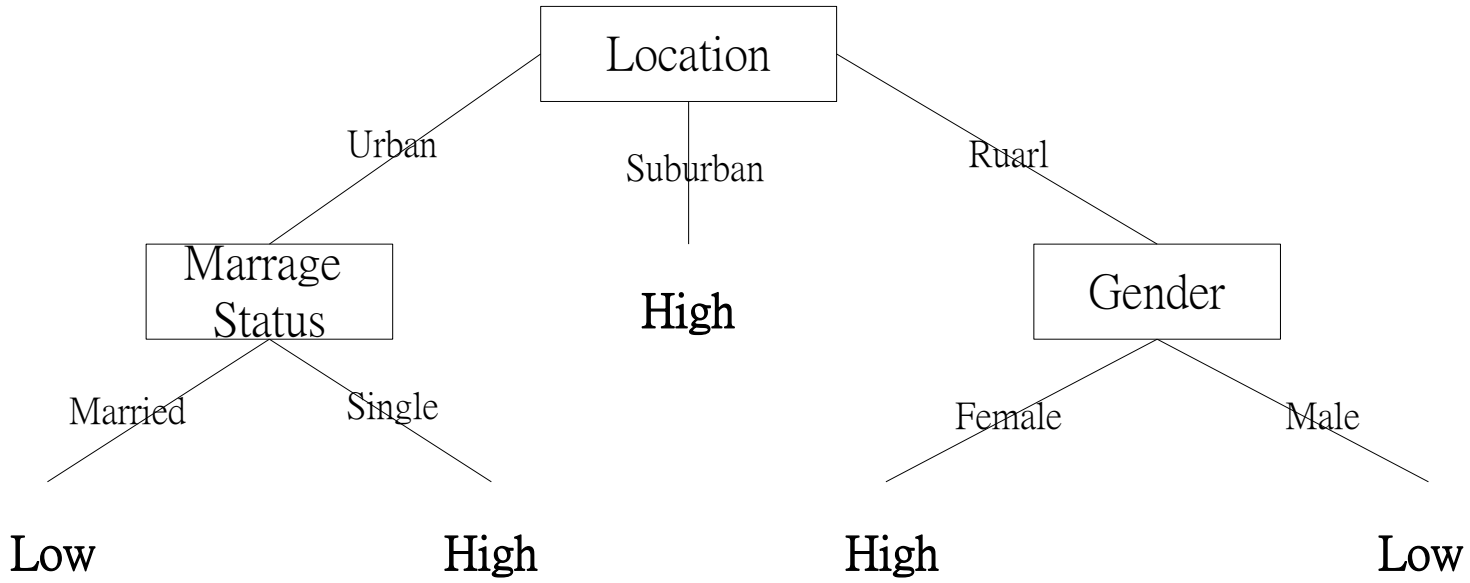
Sample Training Data

No	Attributes				Class
	Location	Age	Marriage status	Gender	Low
1	Urban	Below 21	Married	Female	Low
2	Urban	Below 21	Married	Male	Low
3	Suburban	Below 21	Married	Female	High
4	Rural	Between 21 and 30	Married	Female	High
5	Rural	Above 30	Single	Female	High
6	Rural	Above 30	Single	Male	Low
7	Suburban	Above 30	Single	Male	High
8	Urban	Between 21 and 30	Married	Female	Low
9	Urban	Above 30	Single	Female	High
10	Rural	Between 21 and 30	Single	Female	High
11	Urban	Between 21 and 30	Single	Male	High
12	Suburban	Between 21 and 30	Married	Male	High
13	Suburban	Below 21	Single	Female	High
14	Rural	Between 21 and 30	Married	Male	Low

A Complex Decision Tree

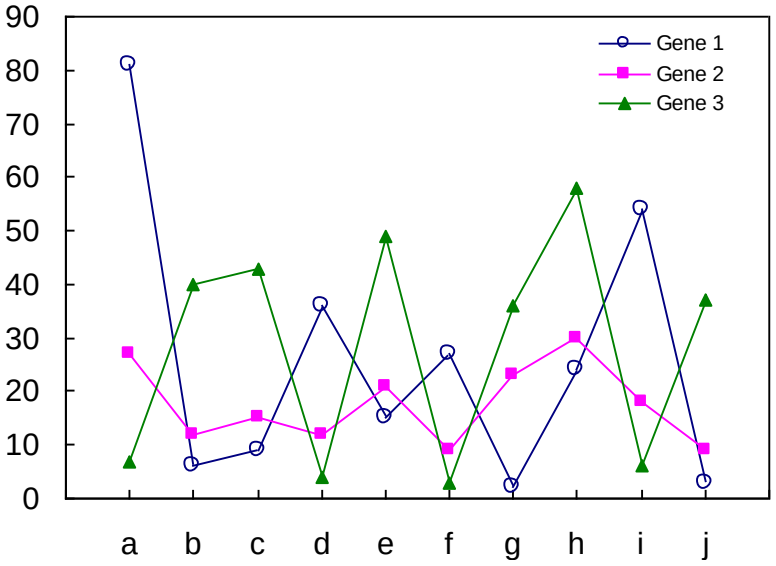


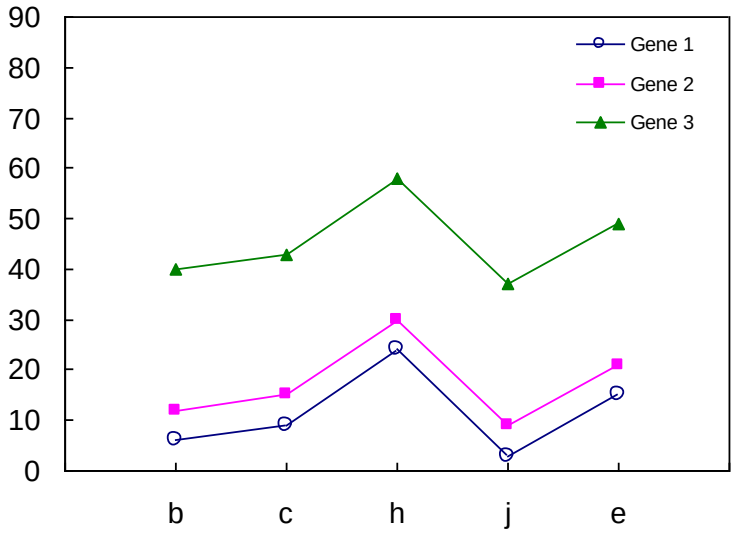
A Compact Decision Tree

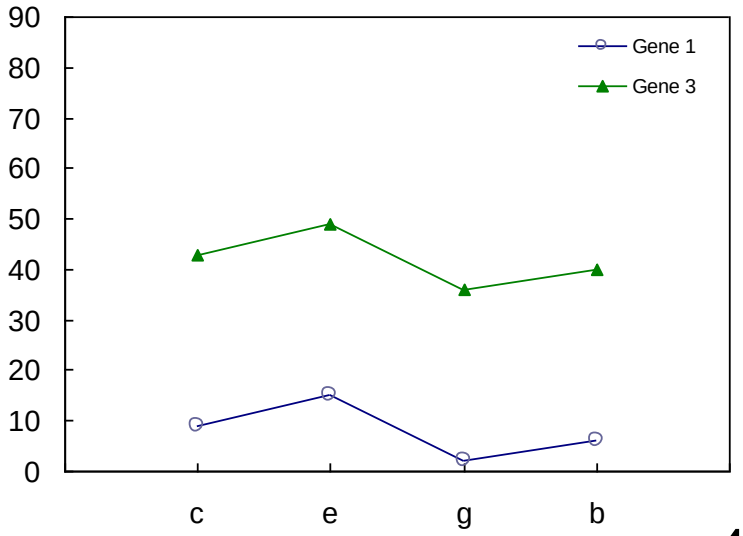


Its predictive power is often higher than that of a complex decision tree.

Subspace Clustering





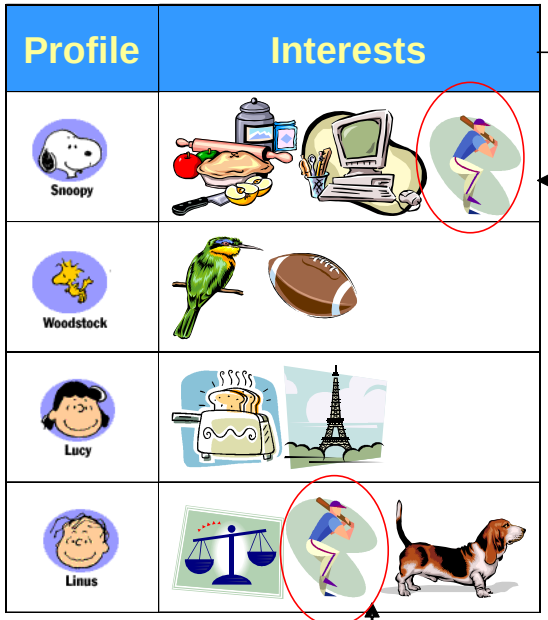


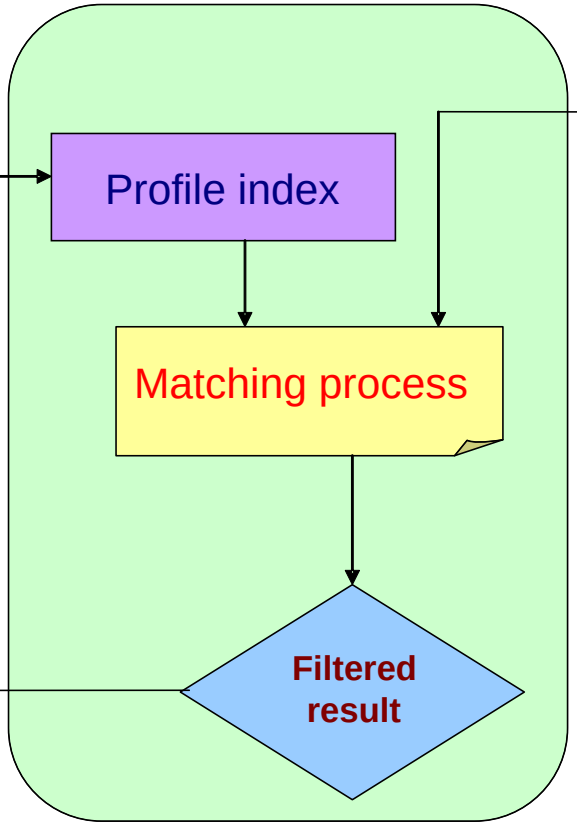
Subspace Cluster :

{gene1, gene2, gene3} x {b, c, h, j, e}

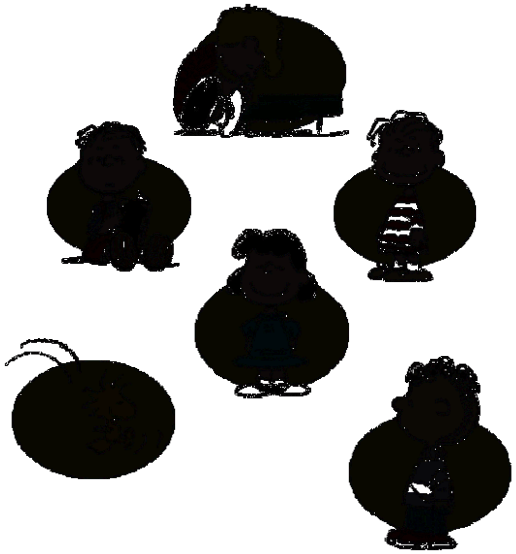
{gene1, gene3} x {c, e, g, b}

Web DB



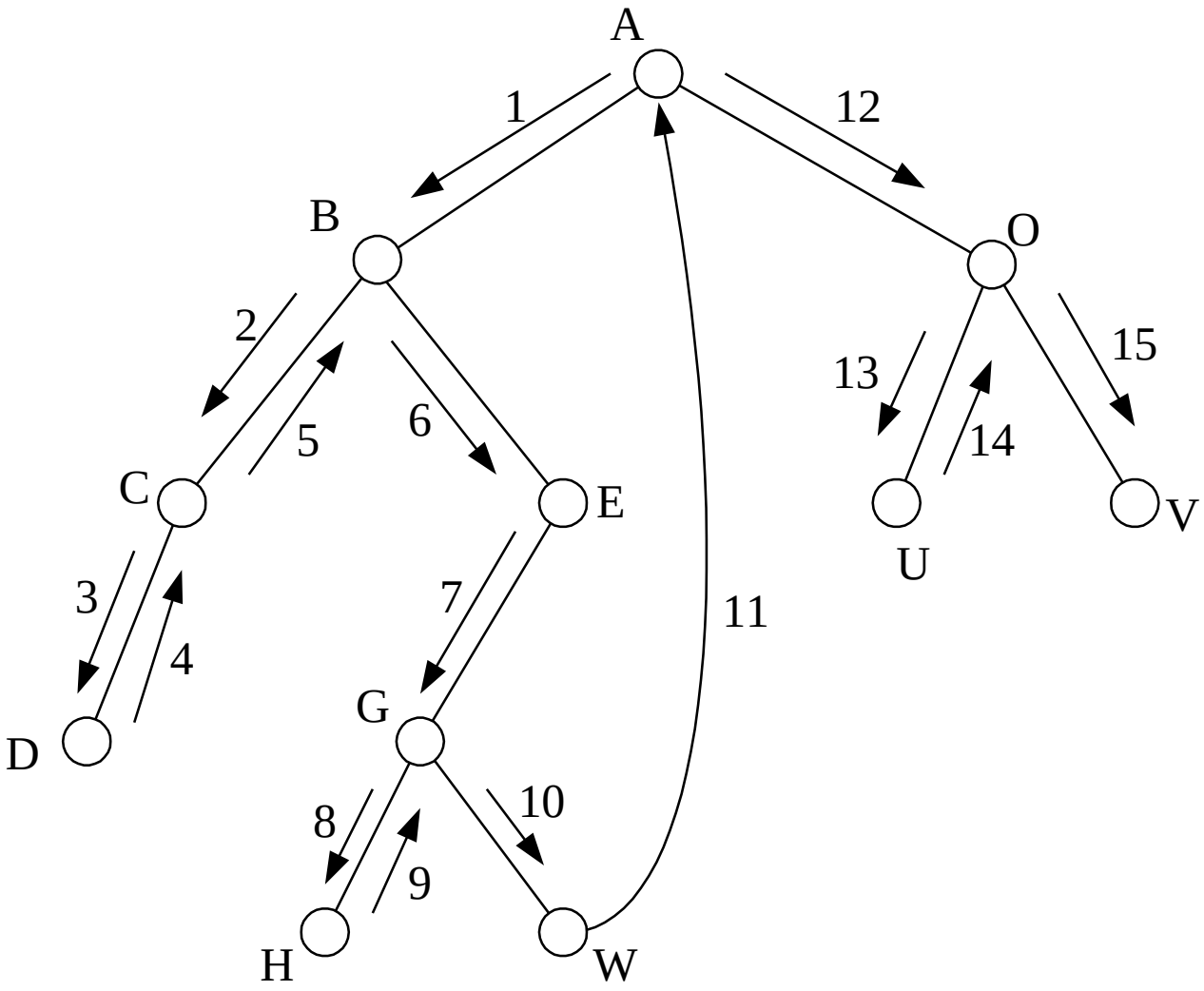


Web Pages



Recommend the page which introduces "basketball" to those people whose interest is "basketball".

Web Mining



An illustrative example for traversal patterns

Data Stream Mining

從封包的Stream Data中找出DOS攻擊的IP

Time	Range	Source	Destination	Protocol	Action
15:54:37	DAEGU CITY HALL	210.101.134.110:14764	192.168.1.40:4500	TCP	Blocked
15:54:49	DAEGU CITY HALL	210.101.134.110:14764	192.168.1.40:4362	TCP	Blocked
15:54:53	DAEGU CITY HALL	210.101.134.110:14764	192.168.1.40:4320	TCP	Blocked
15:55:05	DAEGU CITY HALL	210.101.134.110:14764	192.168.1.40:4500	TCP	Blocked
15:55:17	DAEGU CITY HALL	210.101.134.110:14764	192.168.1.40:4362	TCP	Blocked
15:55:21	DAEGU CITY HALL	210.101.134.110:14764	192.168.1.40:4320	TCP	Blocked
15:55:33	DAEGU CITY HALL	210.101.134.110:14764	192.168.1.40:4500	TCP	Blocked
15:55:45	DAEGU CITY HALL	210.101.134.110:14764	192.168.1.40:4362	TCP	Blocked
15:55:49	DAEGU CITY HALL	210.101.134.110:14764	192.168.1.40:4320	TCP	Blocked
15:56:01	DAEGU CITY HALL	210.101.134.110:14764	192.168.1.40:4500	TCP	Blocked
15:56:13	DAEGU CITY HALL	210.101.134.110:14764	192.168.1.40:4362	TCP	Blocked
15:56:17	DAEGU CITY HALL	210.101.134.110:14764	192.168.1.40:4320	TCP	Blocked
15:56:29	DAEGU CITY HALL	210.101.134.110:14764	192.168.1.40:4500	TCP	Blocked
15:56:41	DAEGU CITY HALL	210.101.134.110:14764	192.168.1.40:4362	TCP	Blocked
15:56:45	DAEGU CITY HALL	210.101.134.110:14764	192.168.1.40:4320	TCP	Blocked
15:56:57	DAEGU CITY HALL	210.101.134.110:14764	192.168.1.40:4500	TCP	Blocked
15:57:09	DAEGU CITY HALL	210.101.134.110:14764	192.168.1.40:4362	TCP	Blocked
15:57:13	DAEGU CITY HALL	210.101.134.110:14764	192.168.1.40:4320	TCP	Blocked
15:57:25	DAEGU CITY HALL	210.101.134.110:14764	192.168.1.40:4500	TCP	Blocked
15:57:37	DAEGU CITY HALL	210.101.134.110:14764	192.168.1.40:4362	TCP	Blocked
15:57:41	DAEGU CITY HALL	210.101.134.110:14764	192.168.1.40:4320	TCP	Blocked

Traditional *vs.* Stream Data

- Traditional Databases
  - Data stored in finite, persistent data sets.
- Stream Data (Big data in cloud)
  - Data as ordered, continuous, rapid, huge amount, time-varying data streams. (In-Memory Databases)

Landmark Window Model

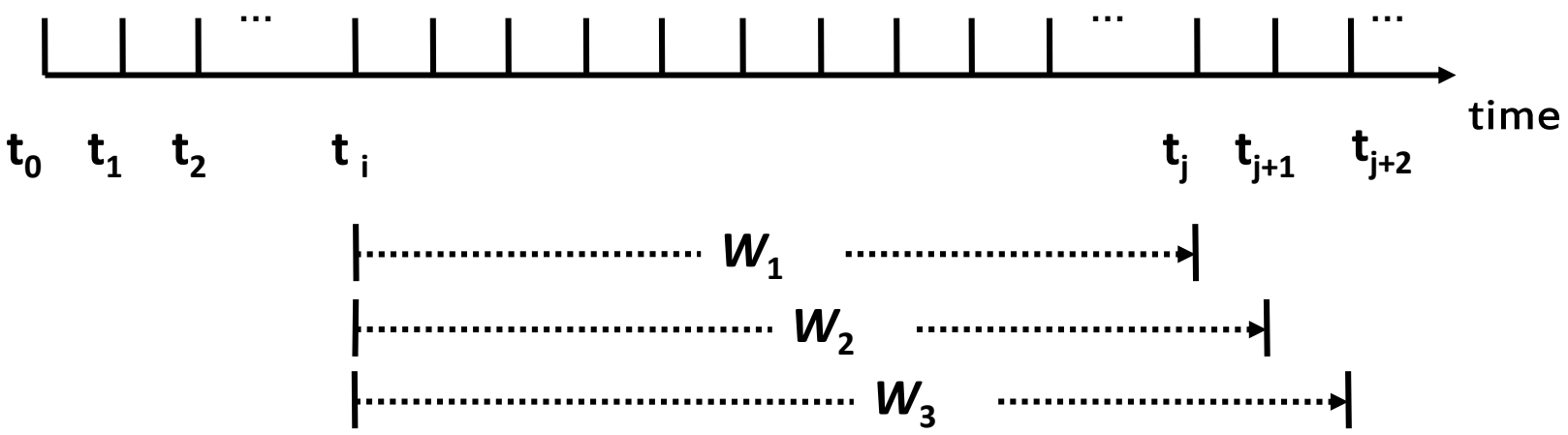


Figure 1. Landmark Window

Titlted-Time Window Model

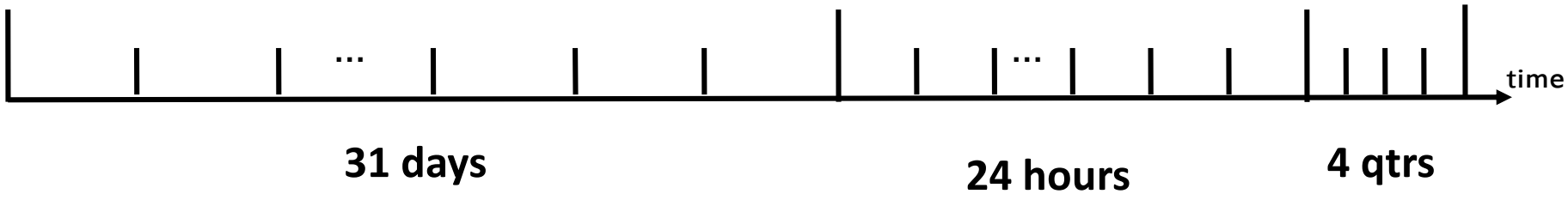


Figure 3. Tilted-Time Window

Sliding Window Model

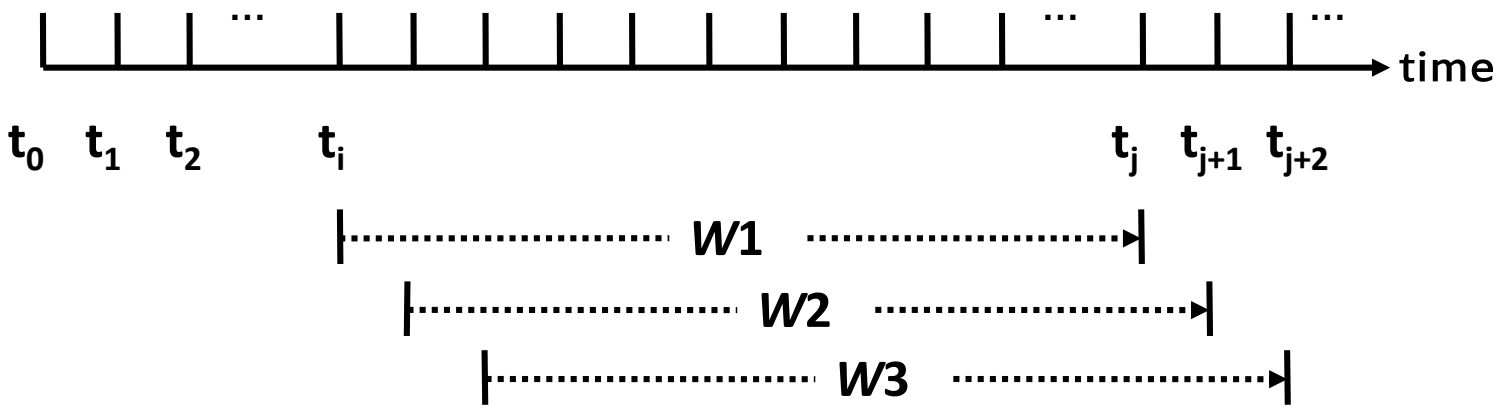
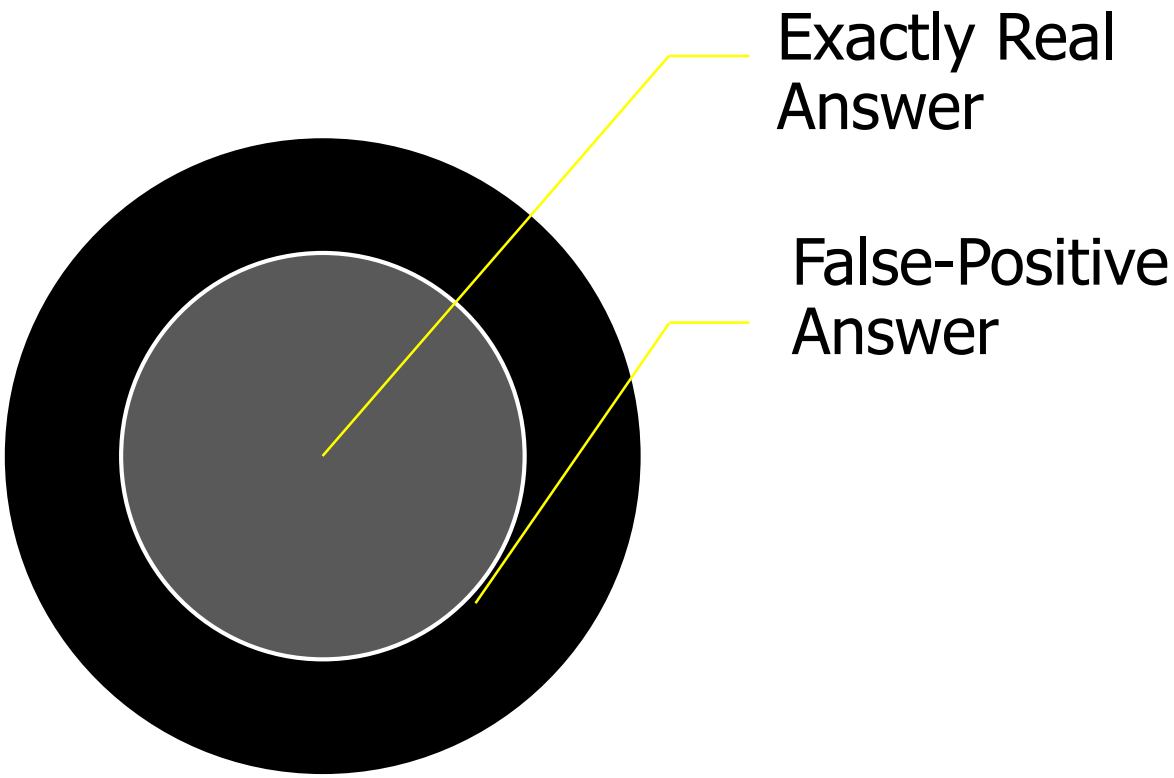
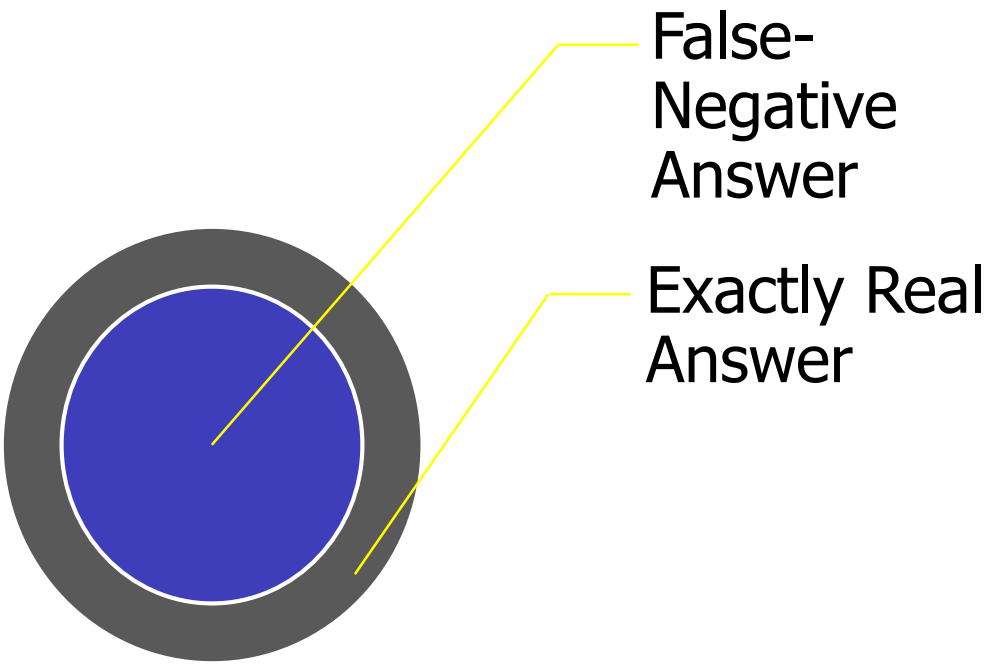


Figure 2. Sliding Window

False-Positive answer



False-Negative answer



Periodicity Mining in Time Series Databases

- Three types of periodic patterns:
  - Symbol periodicity
    - *T* = *abd acb aba abc*
    - Symbol *a* , *p* = 3, *stPos* = 0
  - Sequence periodicity (partial periodic patterns)
    - *T* = *bbaa abbd abca abbc abcd*
    - *Sequence ab, p* = 4, *stPos* = 4
  - Segment periodicity (full-cycle periodicity)
    - *T* = *abcab abcab abcab*
    - *Segement abcab, p* = 5, *stPos* = 0

Mining Frequent Periodic Patterns

How to earn money?

User wants to know whether the pattern periodic or not in the time-series database.

Use computer analyzes time-series database.

Find frequent periodic patterns and predict the future tend of the time-series database.

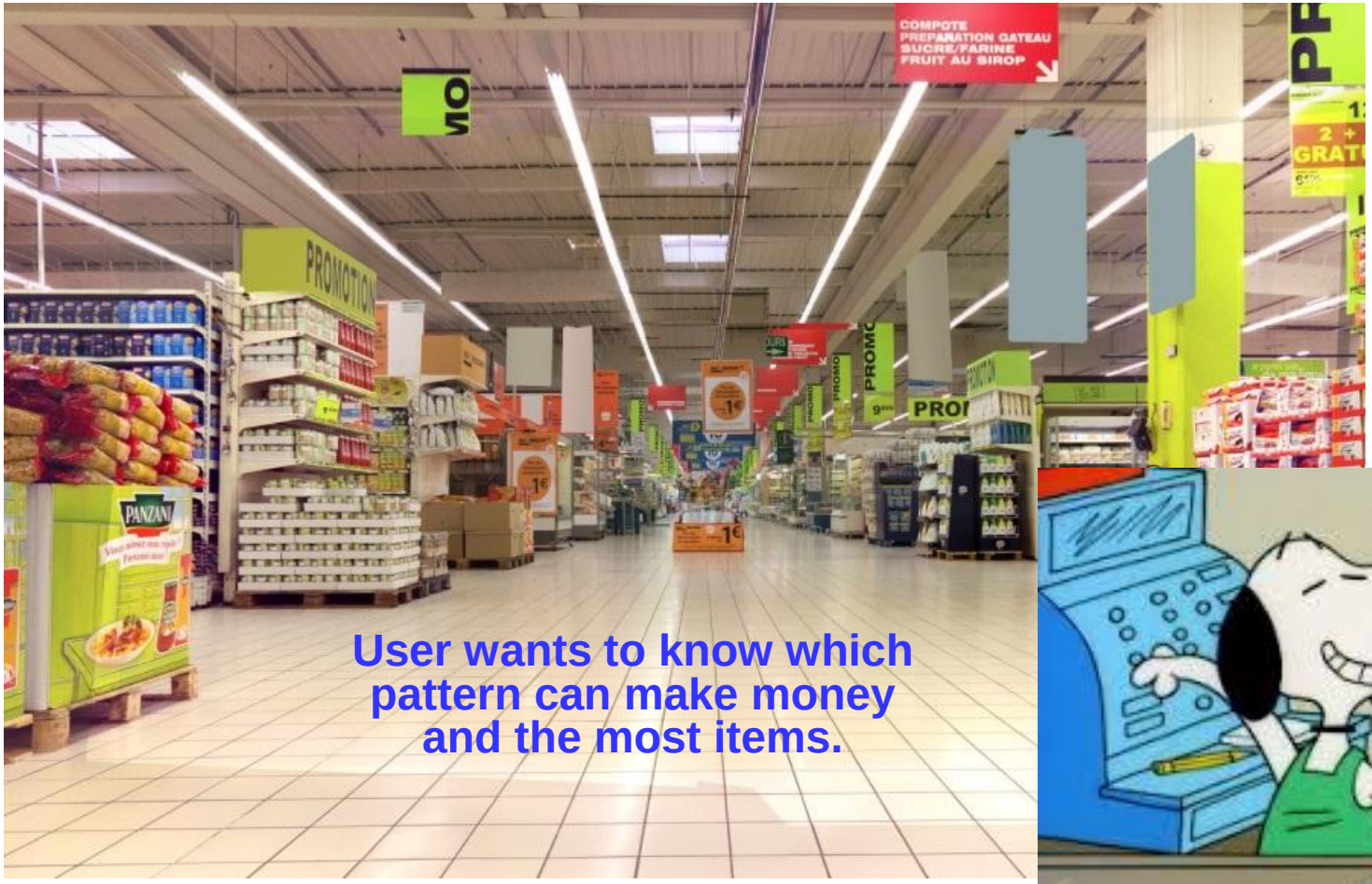
Mining Time-Interval Sequential Patterns

Customers buy something, storage item and time-interval.

Use computer analyzes database.

Find Time-interval patterns not only reveals the order of items but also the time intervals between successive items.

Mining Weight Maximal Frequent Patterns

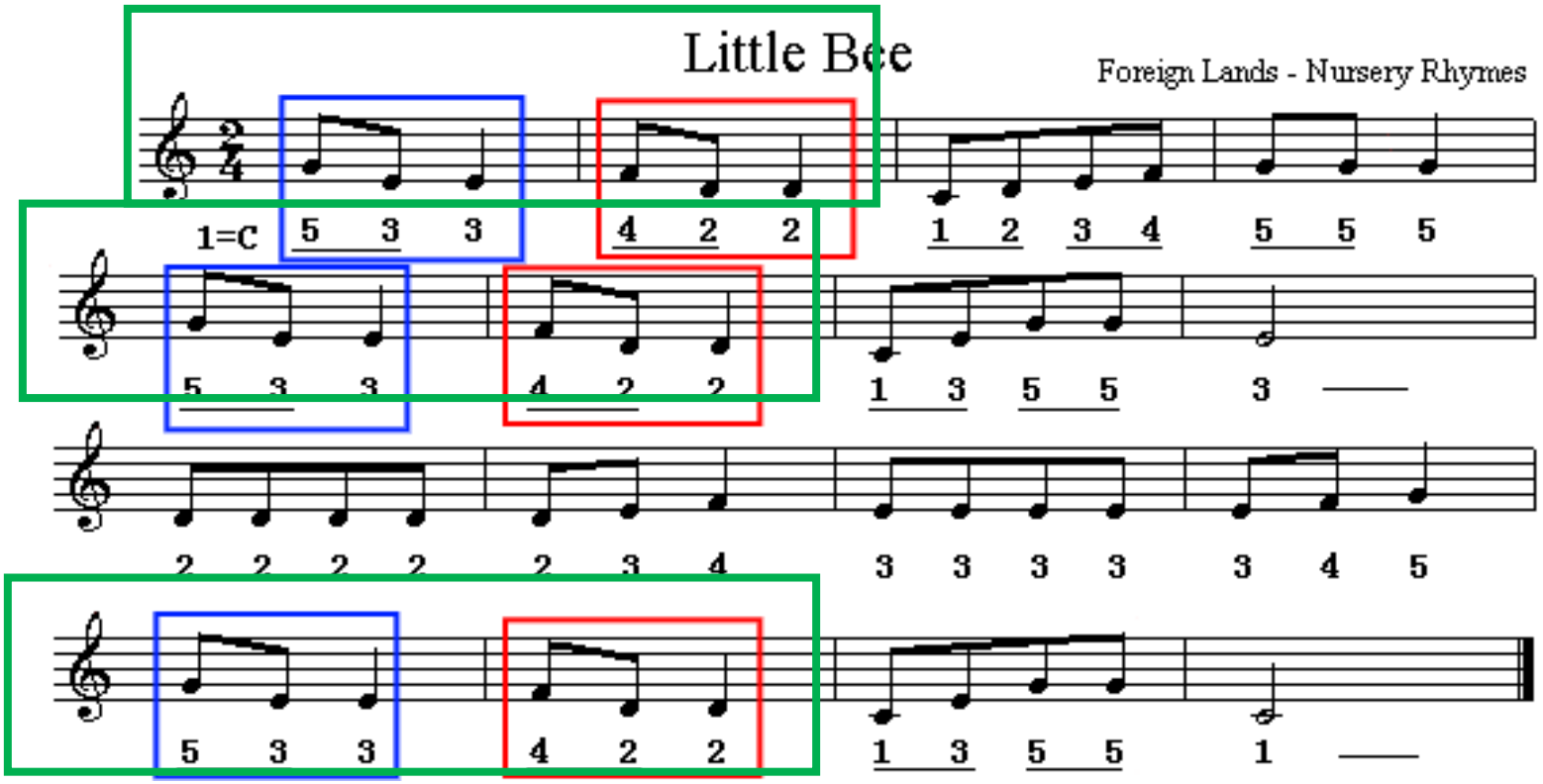


User wants to know which pattern can make money and the most items.

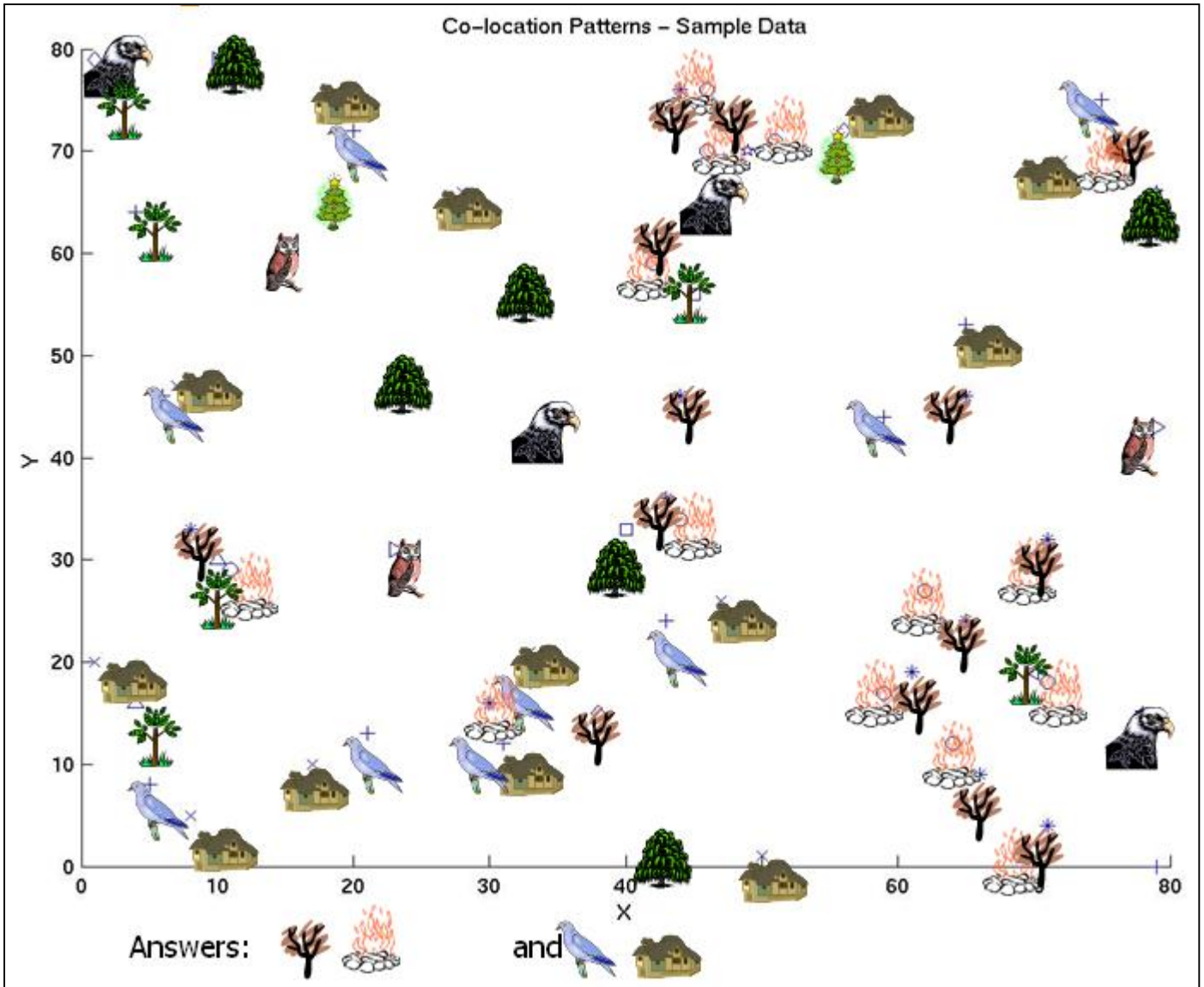
Mining High Utility Patterns

Which itemset can contribute the most profit value of all the transactions?

Monomg Repeating Patterns in Music Databases



Co-Location Patterns

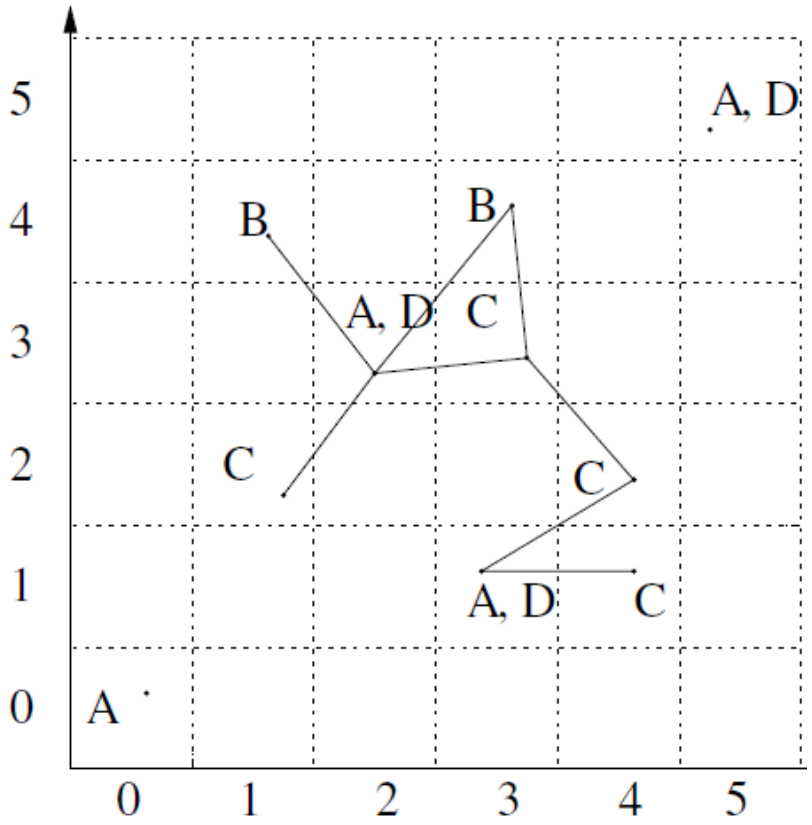


Mining Spatial Co-Location Patterns

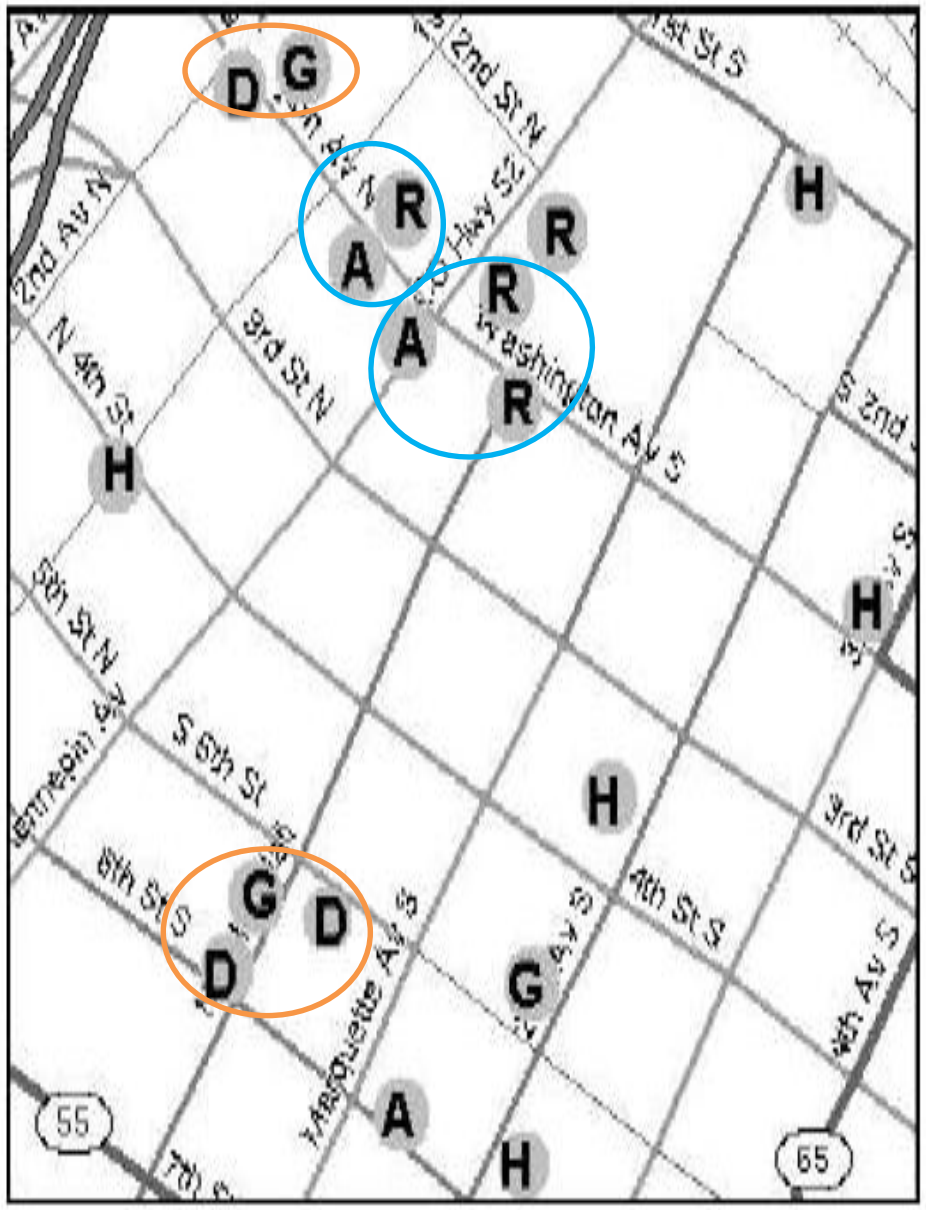
- Ex.

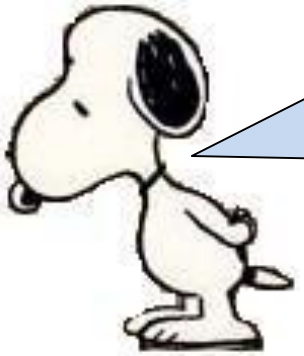
  {A,C}

  {(3,1),(4,1)}
  {(2,3),(1,2)}
  {(2,3),(3,3)}



Co-Location Patterns





Where is good location for retailers to open an after-market ?

A = Auto dealers

R = auto Repair shops

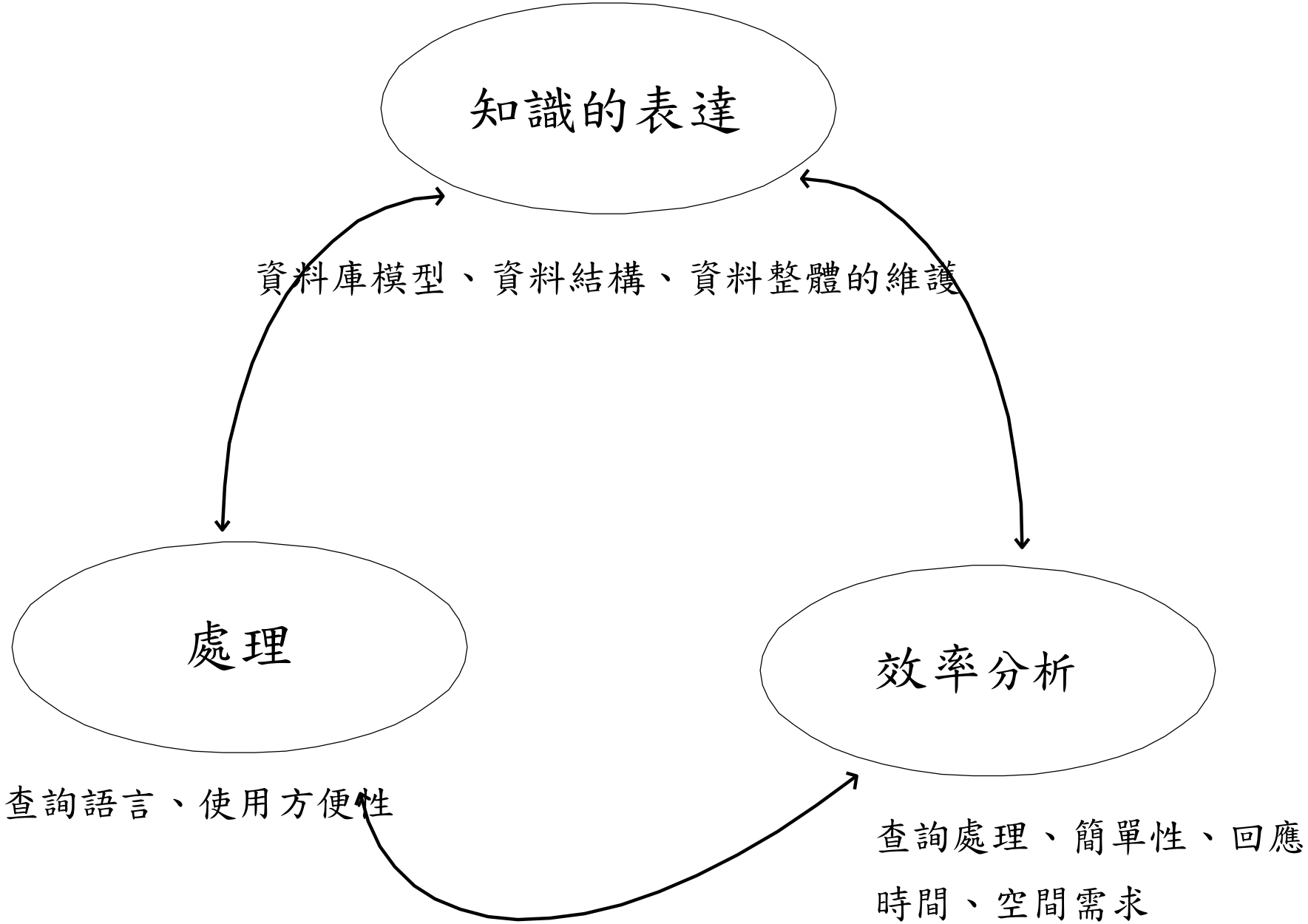
D = Department stores

G = Gift stores

H = Hotels

Co-location patterns:

{A, R}, {D, G}



圖例. 資料庫系統的研究領域