資料探勘:
概念與方法

— 第四章 —

第四章:資料方塊計算與資料產生

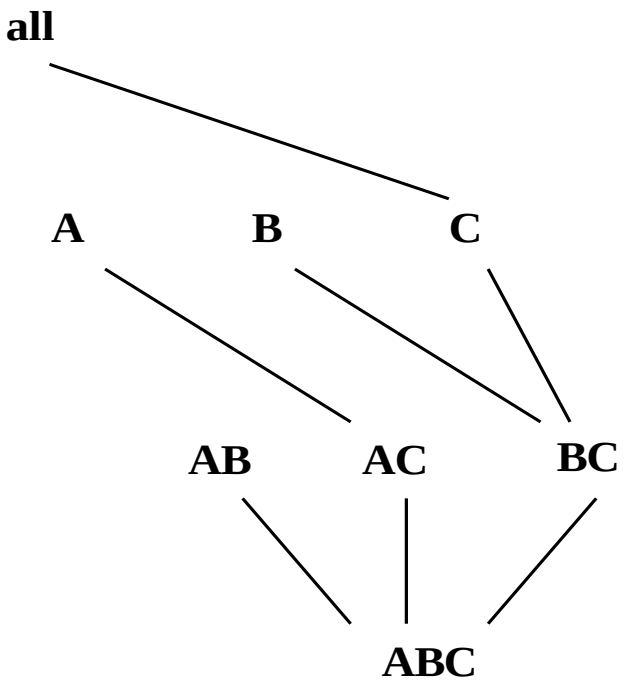
- 有效率計算資料方塊
- 多維度資料庫探索與發掘
- 另外資料一般化的方法-屬性導向歸納

有效率的資料方塊計算

- 計算完整/冰山方塊: 3 方法
  - 由上而下:多向陣列聚合 (Zhao, Deshpande & Naughton, SIGMOD'97)
  - 由下而上:
    - 由下而上計算: BUC (Beyer & Ramarkrishnan, SIGMOD'99)
  - 整合由上而下與由下而上:
    - Star-cubing方法 (Xin, Han, Li & Wah: VLDB'03)
- 計算外殼片段
- 複雜度量方塊計算

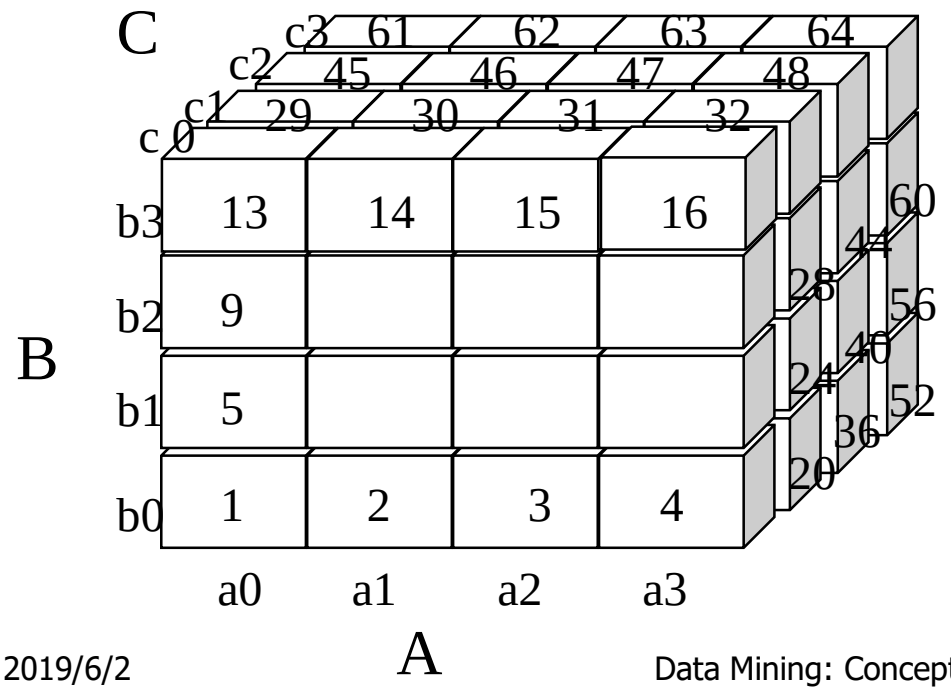
多向陣列聚合

- 陣列式,由下而上的方法
- 使用多維度小塊
- 值組不能直接比較
- 在多維度進行同時聚合
- 中間聚合值被用於計算祖先方塊
- 不能執行*Apriori* 刪除: 沒有冰山最佳化



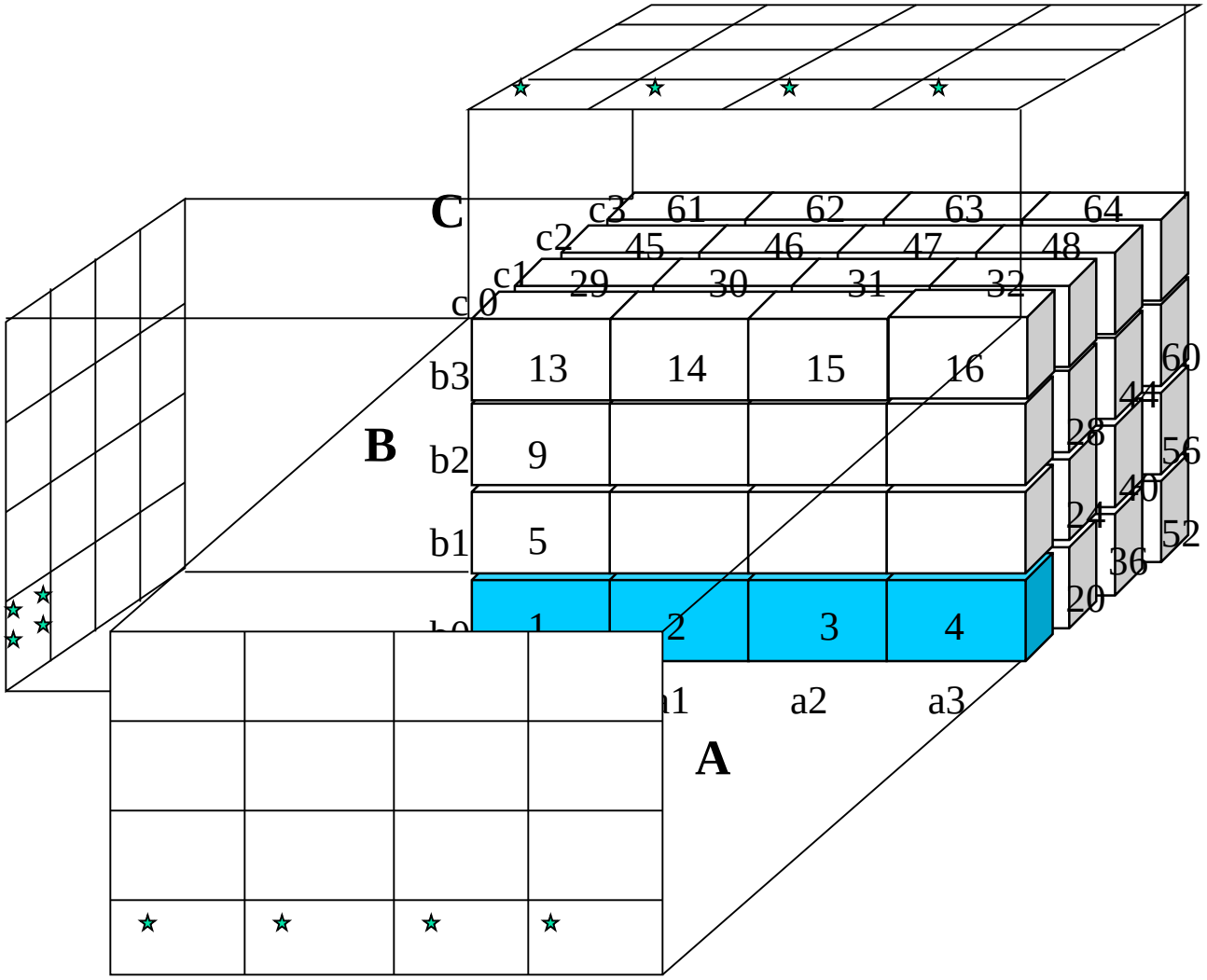
多向陣列聚合的方塊計算 (MOLAP)

- 將陣列分割成小塊 (小到足以放入記憶體進行方塊計算).
- 壓縮稀疏陣列結構的儲存格定址方式：(chunk_id, offset)
- 透過方塊儲存格的存取來計算聚合。儲存格除存取的次序可以進行最佳化，以便能將儲存格存取的次數降到最低 .

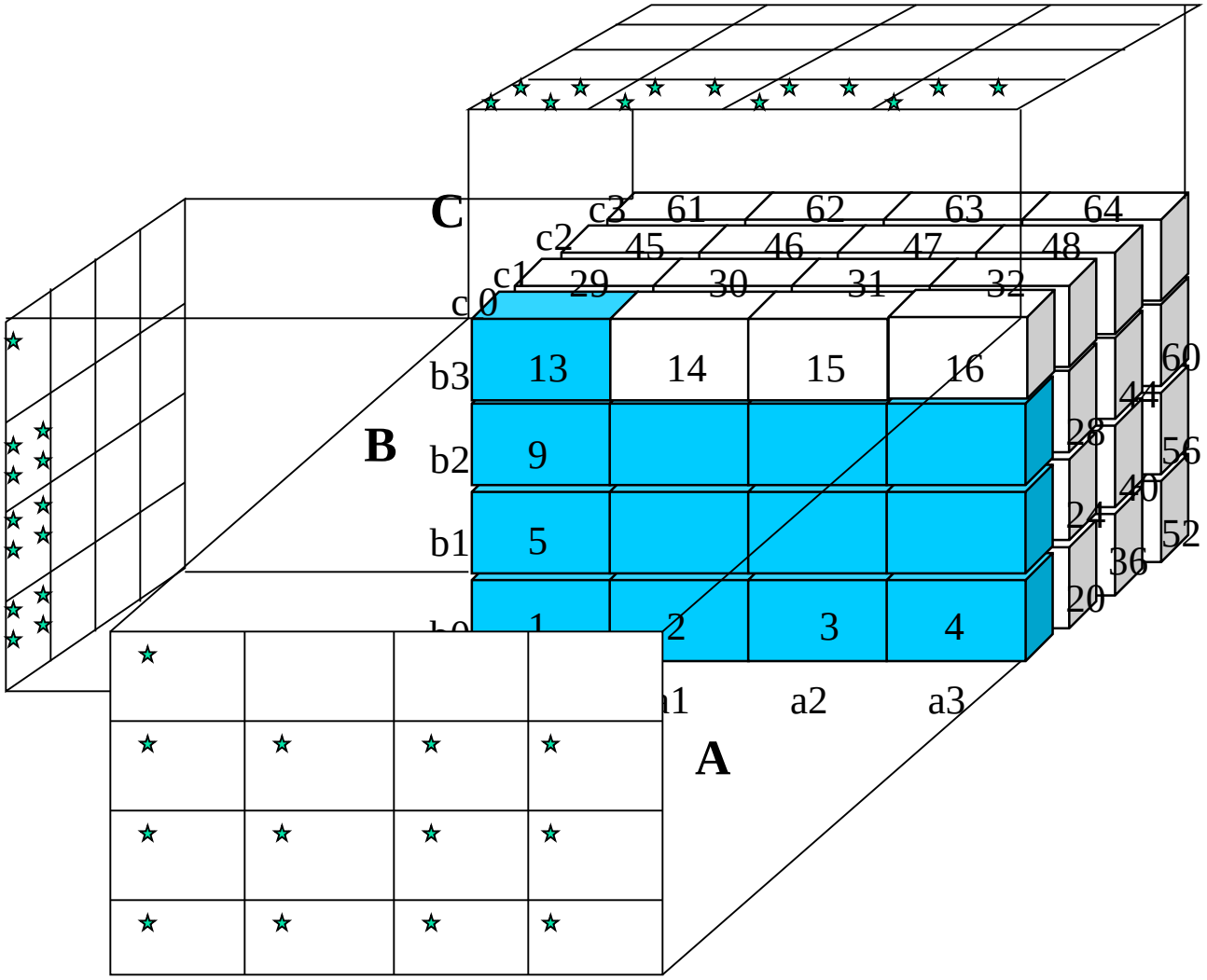


要執行多向陣列聚合,什麼順序是最好的?

多向陣列聚合的方塊計算



多向陣列聚合的方塊計算



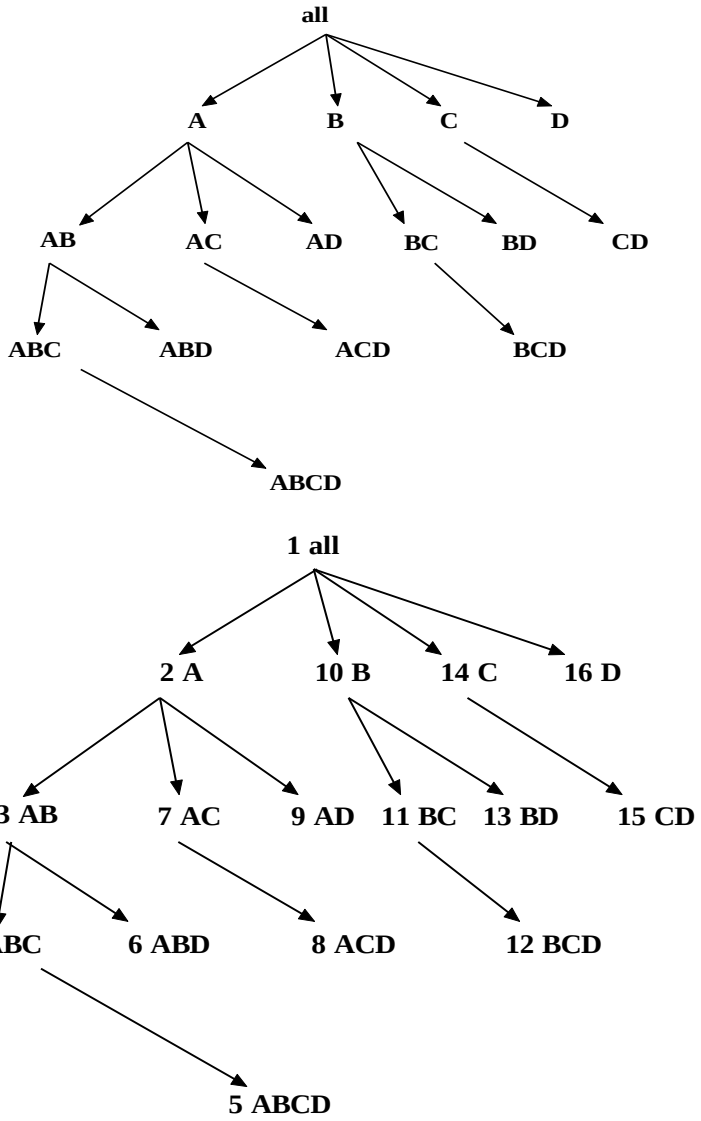
多向陣列聚合的方塊計算(Cont.)

- 方法: 根據平面大小遞增排序來儲存並計算平面
  - 想法: 將最小平面放在記憶體, 每次僅擷取並計算最大平面的一小塊
- 僅適用於相當小的維度

由下而上計算 (BUC)

- BUC (Beyer & Ramakrishnan, SIGMOD'99)
- 由下而上方塊計算

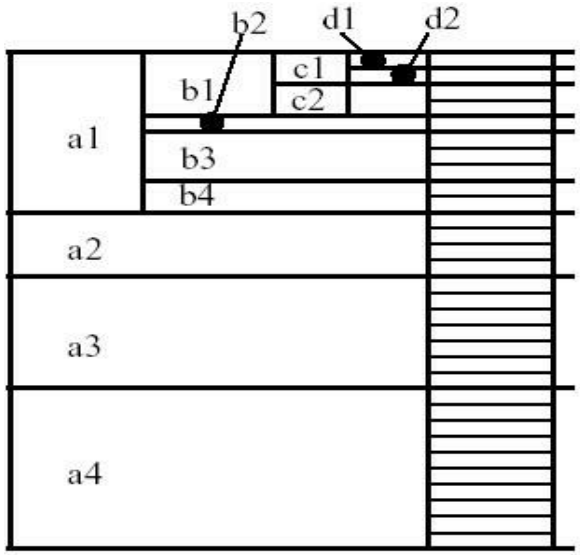
  (請注意計算由頂點長方體開始 !)
- 將維度進行分割並加速冰山刪除
  - 如果分割不滿足最小支持度, 它所有的後裔都可刪除
  - 如果最小支持度 = 1 ⇒ 計算完整方塊
- 沒有同時聚集



BUC: 分割

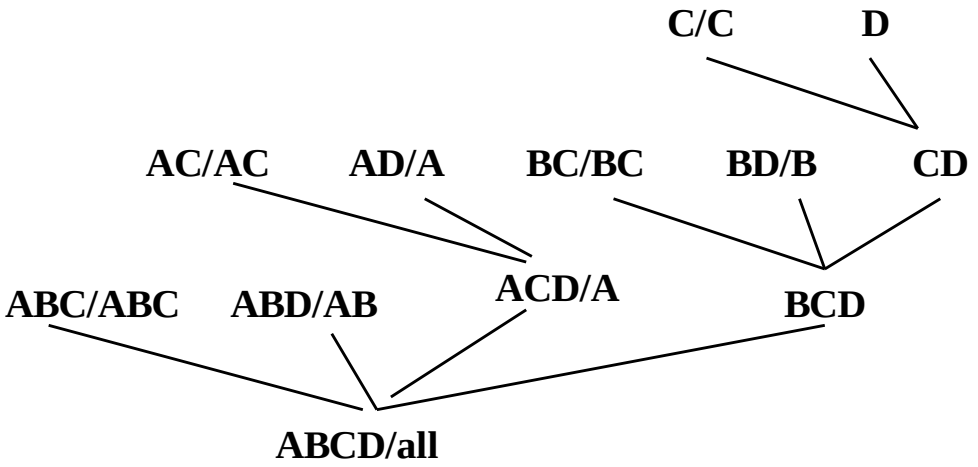
圖 4.7 顯示根據維度 A、B、C、D 不同屬性的分割。BUC 首先對儲存格 cell $(*, *, *, *)$ 也就是 all 進行聚合。接下來維度 A 分成四個分割，每個分割的值組個數儲存於 *dataCount*。

在檢查冰山條件時，BUC 套用 Apriori 特性來節省時間，對維度 A 的 a_1 分割 cell $(a_1, *, *, *)$ 進行聚合。如果 cell $(a_1, *, *, *)$ 滿足最小支持，則對 a_1 分割進行遞迴。在遞迴中 BUC 對 a_1 分割由維度 B 開始，它會檢查 $(a_1, b_1, *, *)$ 是否滿足最小支持，如果它滿足最小支持，它會產生 AB 群組的聚合，並繼續執行遞迴，遞迴會由維度 C 開始。假設 $(a_1, b_1, c_1, *)$ 的個數為 2，因為它不滿足最小支持，根據 Apriori 特性，BUC 不會繼續往下一個維度進行分割，因此 BUC 會刪除 $(a_1, b_1, c_1, *)$ 的探索，然後繼續對 $(a_1, b_1, c_2, *)$ 進行測試。透過在進行遞迴前檢查冰山條件，BUC 可以節省很多處理時間。



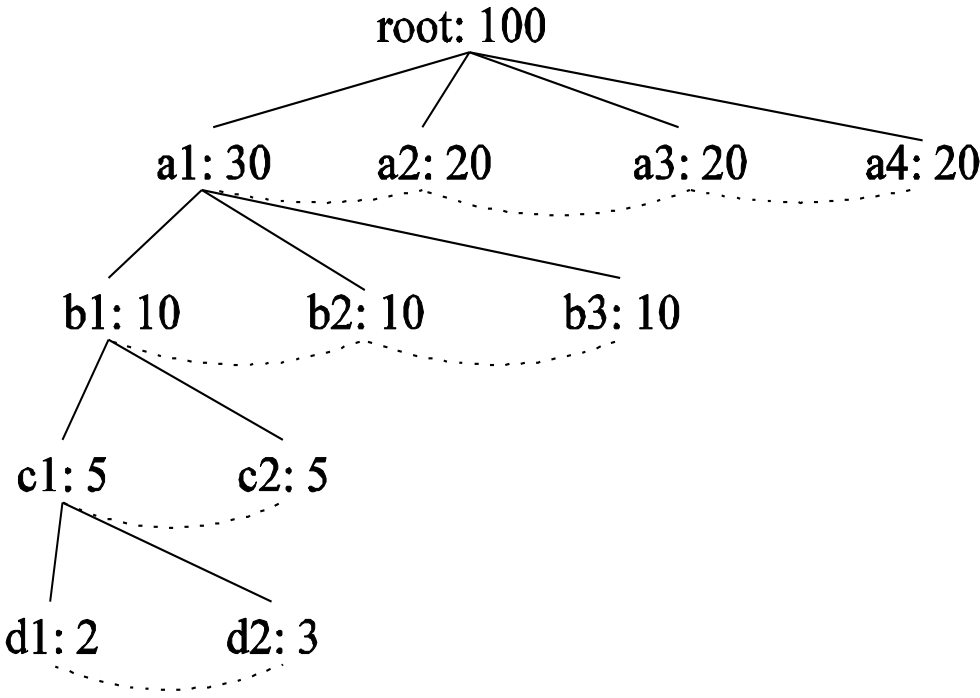
Star-Cubing: 整合方法

- 整合由上而下與由下而上的計算
- ***探索共享維度***
  - 例., 維度 A 為ACD與AD的共享維度
  - ABD/AB 代表方塊 ABD 有共享維度 AB
- 允許分享計算
  - 例.,長方體AB在ABD中已經計算過
- 一個共享維度的聚合值不滿足冰山條件，則它所有的後裔也不會滿足冰山條件
- 共享維度由下而上擴展



星形樹

- 使用樹狀圖代表長方體
- 透過擠壓共同前置路徑來節省空間
- 在每個結點紀錄個數
- 擷取一個特定值組必須橫跨樹



星形屬性與星形點

- 如果單一維度的屬性值不滿足冰山條件，要對這樣的節點進行冰山計算是無益處的
  - 例., b_2, b_3, b_4, c_1, c_2, c_4, d_1, d_2, d_3
- 如果在屬性的節點單一維度的聚合值不滿足冰山條件，我們稱屬性的節點為星節點

A	B	C	D	Count
a1	b1	c1	d1	1
a1	b1	c4	d3	1
a1	b2	c2	d2	1
a2	b3	c3	d4	1
a2	b4	c3	d4	1

範例: 星形縮減

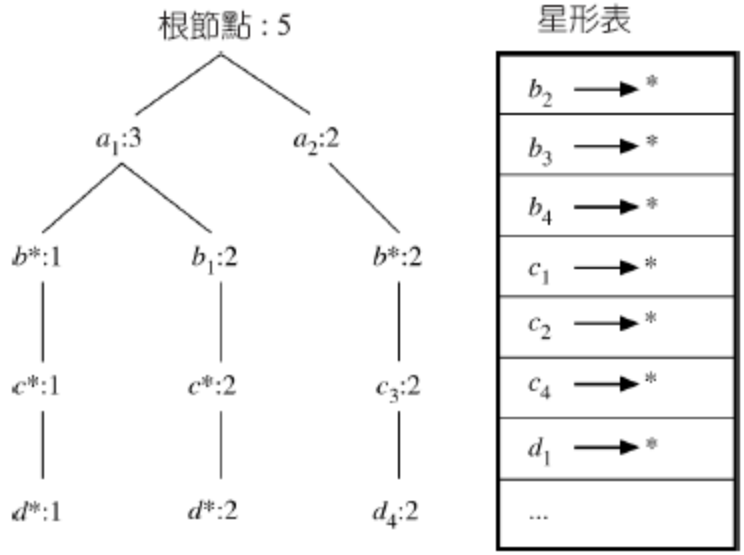
- 假設 minsup = 2
- 執行一維聚合. 將屬性值count < 2 用 *表示, 並將所有*進行壓縮
- 藉由擠壓星形節點，星形樹提供一個對原始資料無誤差的壓縮方法

A	B	C	D	Count
a1	b1	*	*	1
a1	b1	*	*	1
a1	*	*	*	1
a2	*	c3	d4	1
a2	*	c3	d4	1

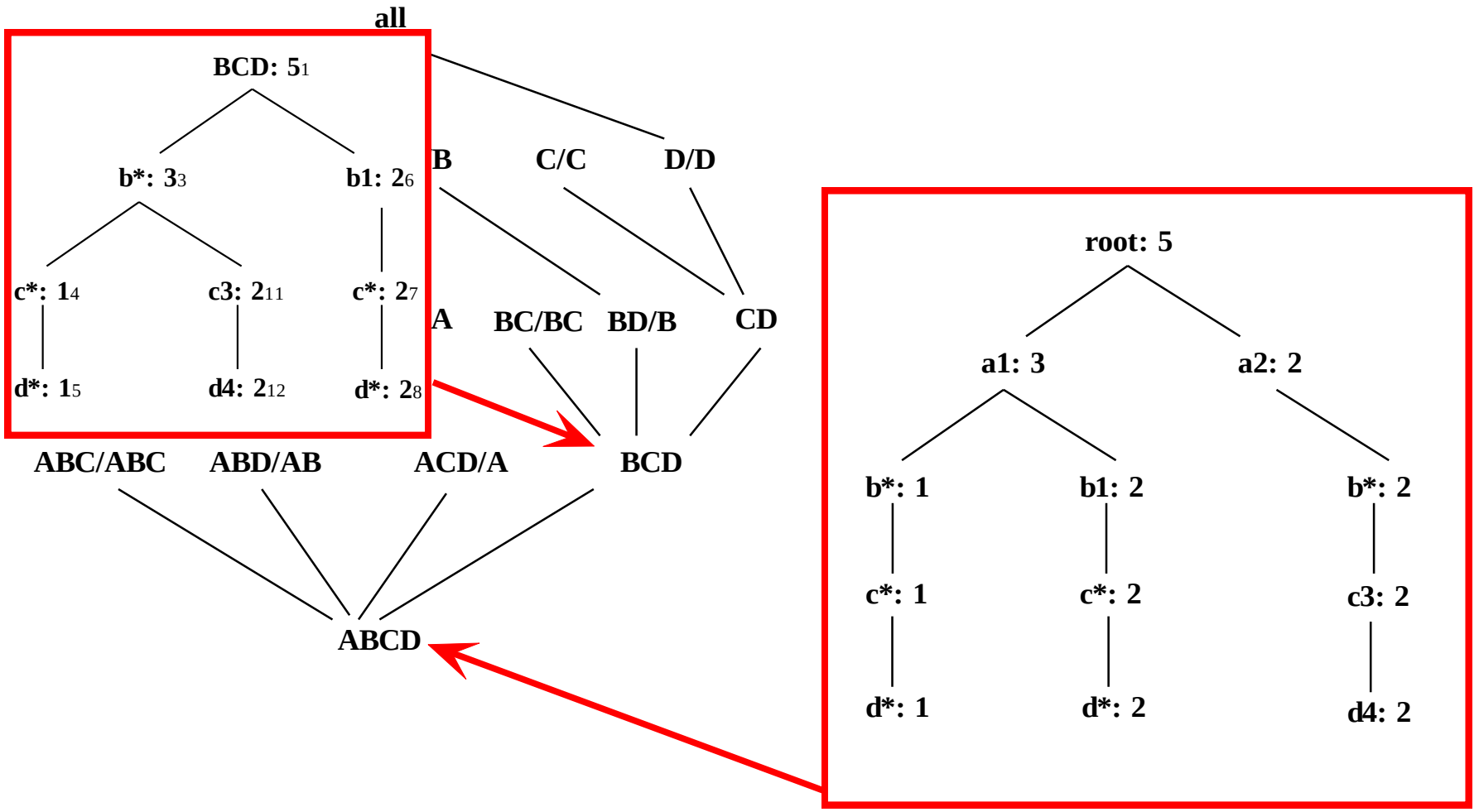
A	B	C	D	Count
a1	b1	*	*	2
a1	*	*	*	1
a2	*	c3	d4	2

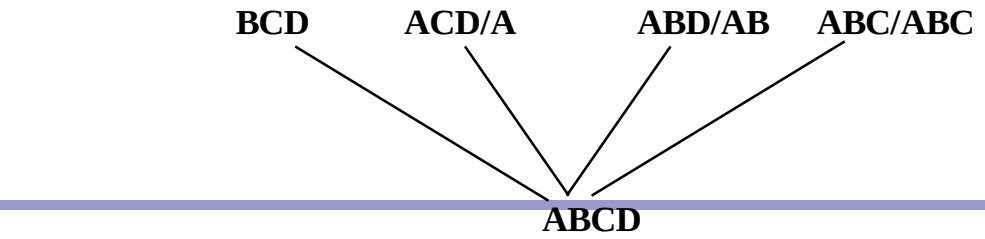
星形樹

- 給定一個壓縮表, 它可以建立一個對應星形樹
- 星形樹提供一個對原始資料無誤差的壓縮方法



Star-Cubing 方法—對晶格數進行DFS





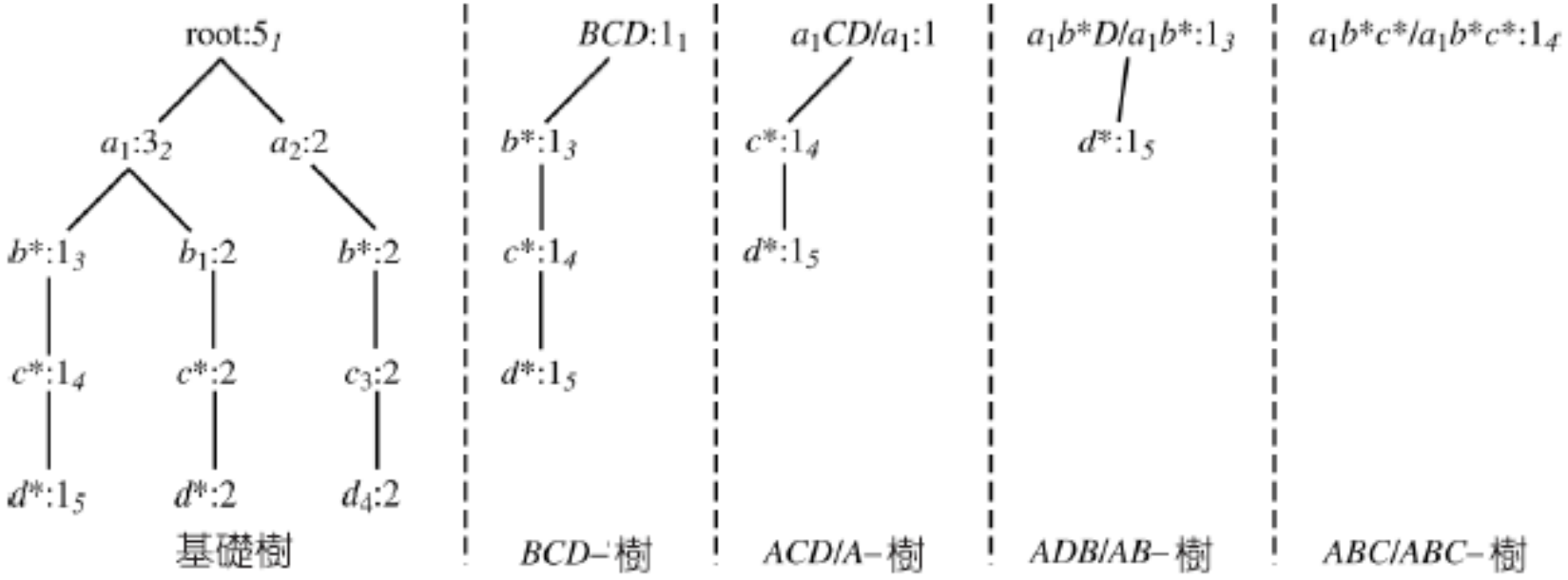
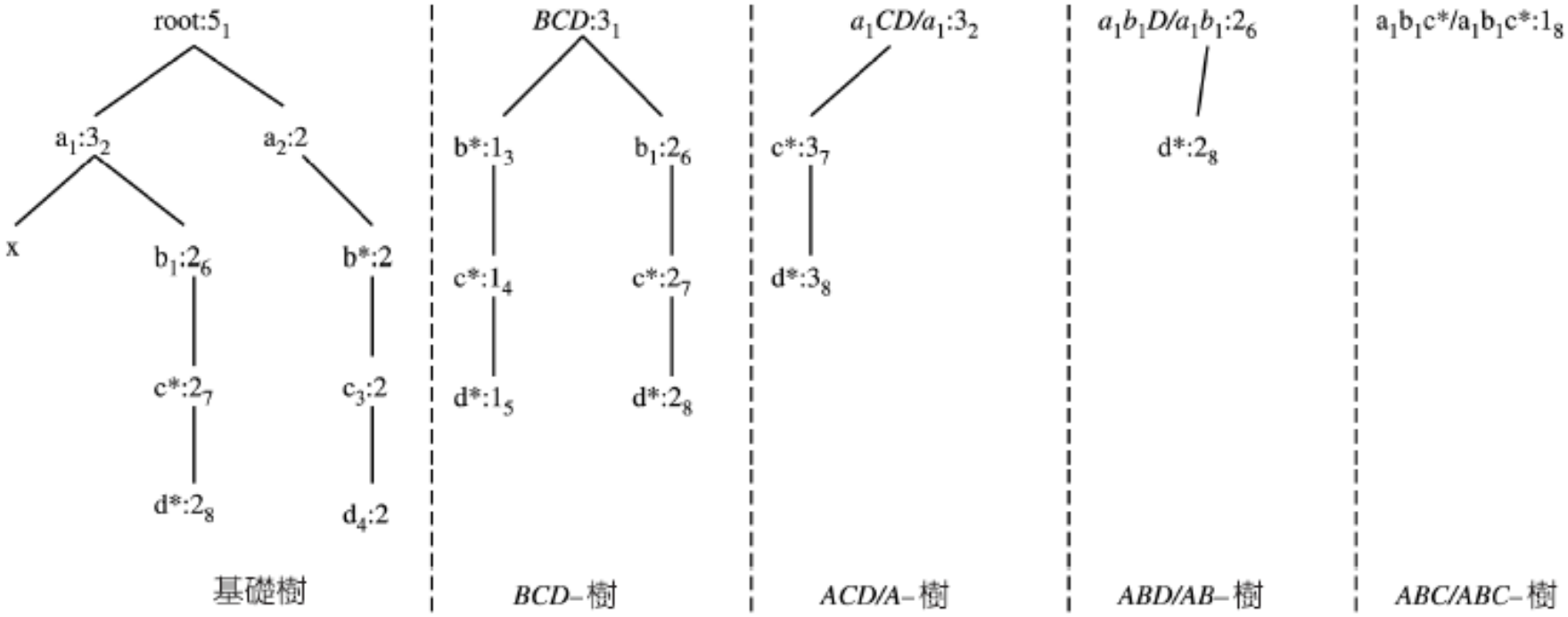


圖 4.11 第一階段聚合：處理基礎樹的最左邊分枝。

Star-Cubing 方法—在星形樹進行DFS



為何高維度OLAP?

- 現今產生方塊方法:
  - 冰山方塊的計算與所需空間仍舊會很大
  - 要決定適當冰山界限是有困難的
  - 冰山方塊無法進行遞增更新
- 高維度OLAP 應用
  - 科學與工程分析
  - 生物資料分析: 數千個基因
  - 統計問卷: 數百個變數

快速高維度OLAP 探勘方塊

- 觀察:大部分OLAP運算一次僅僅對少部分維度進行運算
- 半即時計算模型
  1. 對於一個高維度資料集，我們將維度分割成不相交的維度片段，把每個片段轉換為相對的倒轉索引表示，然後建立外殼片段方塊
  2. 使用事先計算的外殼片段方塊，動態組合並即時計算所需資料方塊長方體的儲存格

範例

- 假設方塊度量為count ()

交易編號	**A**	**B**	**C**	**D**	**E**
1	a1	b1	c1	d1	e1
2	a1	b2	c1	d2	e1
3	a1	b2	c1	d1	e2
4	a2	b1	c1	d1	e2
5	a2	b1	c1	d1	e3

- 五個維度成兩個片段:
  - (A, B, C) 與 (D, E)

1-維倒轉索引

- 建立傳統倒轉索引

屬性值	交易清單編號	清單大小
a1	1 2 3	3
a2	4 5	2
b1	1 4 5	3
b2	2 3	2
c1	1 2 3 4 5	5
d1	1 3 4 5	4
d2	2	1
e1	1 2	2
e2	3 4	2
e3	5	1

外殼片段方塊

- 將1-維倒轉索引通用化至多維度方塊

表4.6 長方體 AB

儲存格	交集	交易編號清單	清單大小
(a_1, b_1)	$\{1, 2, 3\} \cap \{1, 4, 5\}$	$\{1\}$	1
(a_1, b_2)	$\{1, 2, 3\} \cap \{2, 3\}$	$(2, 3\}$	2
(a_2, b_1)	$\{4, 5\} \cap \{1, 4, 5\}$	$\{4, 5\}$	2
(a_2, b_2)	$\{4, 5\} \cap \{2, 3\}$	$\{\ \}$	0

外殼片段方塊 (2)

- 計算資料方塊ABC與DE所有長方體並記錄倒轉索引
- 例, 外殼片段方塊 ABC 包含 7 長方體:
  - A, B, C
  - AB, AC, BC
  - ABC
- 這些可在離線狀態下完成

ID_Measure 表

- 如果度量不為count (), 用*ID_measure* 表來儲存其它度量的計算

交易編號	項目個數	加總
1	5	70
2	3	10
3	8	20
4	5	40
5	2	30

Frag-Shells 方法

運算法則：**Frag-Shells**。對一個高維度基礎表(例如基礎長方體)計算外殼片段。

輸入：一個 n 維度基礎長方體 B，也就是$(A_1, \cdots, A_n)$

輸出：

- 一組片段分割 $\{P_1, \cdots, P_k\}$ 與其相對應(區域)片段方塊 $\{S_1, \cdots, S_k\}$，而 P_i 代表某些維度集合 (S) 並且 $P_1 \cup \cdots \cup P_k$ 會構成所有維度。
- 一組 *ID_measure* 陣列當指標不是值組個數，count()

方法：

(1) 將一組維度$(A_1, \cdots, A_n)$分割成 k 個片段 $P_1, \cdots, P_k$ (根據資料與查詢分佈)
(2) 檢視長方體 B 一次並執行下列步驟 {
(3) 將每個 <*TID, measure*> 插入 *ID_measure* 陣列
(4) **for** 對於每個維度 A_i 的每個屬性值 a_j
(5) 建立一個倒轉索引項目： $<a_j, TIDlist>$
(6) }
(7) **for** 每個片段分割 P_i
(8) 藉由對它們相對應的 TIDlists 進行交集與計算它們的指標來建立一個區域片段方塊 S_i
}

利用非反一致冰山條件建立方塊

- 大部分方塊方法無法有效利用非反一致冰山條件計算方塊
- 例

```
compute cube sales_avg_iceberg as
select 月, 城市, 客户群組, avg(價格), count(*)
from salesInfo
cube by 月, 城市, 客户群組
having avg(價格)>=800 and count(*)>=50
```

- 需要探討如何將限制深植於方塊處理

Non-Anti-Monotonic Iceberg Condition

- 反一致:當某些儲存格不滿足條件時，所有的後裔也不滿足條件
- 方塊查詢 avg 具非反一致!
  - 在R_1區域中的項目，如電視的平均價格小於800元，但是R_1的子區域電視項目的平均價格仍有可能超過800元

前面k個儲存格的平均

- 假設 (*, Van, *) 包含 1,000 紀錄
  - Avg(price) 維這1000銷售平均價格
  - Avg^{50}(price) 為前50銷售平均價格 (前50 是根據銷售價格)
- 前k 平均為反一致
  - 在Van. 具有avg(price) <= 800 的前50銷售→ 在Van. 二月前50銷售一定 avg(price) <= 800

Chapter 4: Data Cube Computation and Data Generalization

- 有效率計算資料方塊
- 多維度資料庫探索與發掘
- 另外資料一般化的方法-屬性導向歸納

資料方塊的發掘導向探索

- 假設導向
  - 透過使用者探索, 龐大搜尋空間
- 發掘導向 (Sarawagi, et al.'98)
  - 有效瀏覽龐大 OLAP 資料方塊
  - 事先計算度量顯示異常資料，並用來引導使用者在所有的聚合層次中進行資料分析
  - 異常:如果資料方塊的儲存格值明顯不同於其他參與儲存格的值
  - 視覺提示如背景顏色用於反應異常層度

異常種類與計算

- 異常指標
  - SelfExp: 相對於同一層其他聚合儲存格的驚訝層度
  - InExp:相對於較低層次儲存格的驚訝層度
  - PathExp:對每個往下鑽探路徑儲存格的驚訝層度
- 異常指標的計算可以與方塊計算重疊
- 異常可以如同預先計算聚合被儲存, 索引與擷取

範例: 發掘導向資料方塊

銷售加總	月											
	一月	二月	三月	四月	五月	六月	七月	八月	九月	十月	十一月	十二月
總計		1%	-1%	0%	1%	3%	-1%	-9%	-1%	2%	-4%	3%

圖4.15 根據時間的銷售差異。

平均銷售	月											
項目	一月	二月	三月	四月	五月	六月	七月	八月	九月	十月	十一月	十二月
Sony b/w printer		9%	-8%	2%	-5%	14%	-4%	0%	41%	-13%	-15%	-11%
Sony color printer		0%	0%	3%	2%	4%	-10%	-13%	0%	4%	-6%	4%
HP b/w printer		-2%	1%	2%	3%	8%	0%	-12%	-9%	3%	-3%	6%
HP color printer		0%	0%	-2%	1%	0%	-1%	-7%	-2%	1%	-4%	1%
IBM desktop computer		1%	-2%	-1%	-1%	3%	3%	-10%	4%	1%	-4%	-1%
IBM laptop computer		0%	0%	-1%	3%	4%	2%	-10%	-2%	0%	-9%	3%
Toshiba desktop computer		-2%	-5%	1%	1%	-1%	1%	5%	-3%	-5%	-1%	-1%
Toshiba laptop computer		1%	0%	3%	0%	-2%	-2%	-5%	3%	2%	-1%	0%
Logitech mouse		3%	-2%	-1%	0%	4%	6%	-11%	2%	1%	-4%	0%
Ergo-way mouse		0%	0%	2%	3%	1%	-2%	-2%	-5%	0%	-5%	8%

4.16 對每個項目-時間組合的銷售差異。

平均銷售	月											
地區	一月	二月	三月	四月	五月	六月	七月	八月	九月	十月	十一月	十二月
北		-1%	-3%	-1%	0%	3%	4%	-7%	1%	0%	-3%	-3%
南		-1%	1%	-9%	6%	-1%	-39%	9%	-34%	4%	1%	7%
東		-1%	-2%	2%	-3%	1%	18%	-2%	11%	-3%	-2%	-1%
西		4%	0%	-1%	-3%	5%	1%	-18%	8%	5%	-8%	1%

圖4.17 對每個區域*IBM*桌上型電腦的銷售差異。

不同層度的複雜聚合：多特性方塊

- 多特性方塊 (Ross, et al. 1998):在不同層度下對多個獨立聚合進行複雜查詢
- 例.對{項目, 區域, 月}所有的子集合進行群組，尋找每個群組2004年最大價格

```
select      項目, 區域, 月, max(價格), sum(R.銷售)
from        購買
where       年 = 2004
cube by     項目, 區域, 月:R
such that   R.價格 = max(價格)
```

- 對{項目, 區域, 月}所有的子集合進行群組，尋找每個群組2004年最大價格，在具有最大價格的值組找出項目最小與最大的架上時間, 並且在具有最大價格的值組中，找出最小架上時間商品的總銷售額，與最大架上時間商品的總銷售額

方塊梯度 (方塊坡度)

- 對多維度空間分析複雜度量的變化
  - 查詢:相對於2003年，2004年在溫哥華平均房價的變化
  - 回答:銷售給在West End專業人士的平均房價跌20%，銷售給市中心商業人士的平均房價上漲10%等
- Imielinski et al提出方塊坡度.
  - 維度改變 → 度量改變
  - 向下鑽探,向上鑽探, 與突變

從方塊坡度到限制多維度梯度分析

- 比關聯規則更具表達性
  - 在使用者設定度量中能找出趨勢
- 挑戰
  - 顯著性限制→限制我們僅檢視某些統計顯著的儲存格
  - 探測限制 →可以讓我們設定有興趣的儲存格
  - 梯度限制→只對儲存格間某些變化感興趣

Chapter 4: Data Cube Computation and Data Generalization

- 有效率計算資料方塊
- 多維度資料庫探索與發掘
- 另外資料一般化的方法-屬性導向歸納

何謂概念描述?

- 概念描述與預測性資料探勘
  - 描述探勘:用簡潔的概念來描述一般抽象層次
  - 預測探勘: 根據資料分析計例模型並預測未知資料趨勢與特性
- 概念描述:
  - 特徵化:提供特定資料精簡的彙總
  - 比較化:提供兩組資料體比對的描述

屬性導向歸納

- 1989提出 (KDD '89 workshop)
- 不受限類別資料或特定度量
- 想法?
  - 使用資料庫查詢收集工作相關的資料
  - 在相關的資料集中檢視屬性的不同值並進行一般化。一般化是透過屬性移除或屬性一般化的動作
  - 對相同的一般值組進行合併，並對相關的值進行累計
  - 透過不同的形式如圖表或規則顯示給使用者看

屬性導向歸納執行方式

- 資料聚焦:收集資料探勘查詢所需資料，資料探勘查詢通常只與部分資料庫的資料有關
- 屬性移除:如果起始工作關係中一個屬性的屬性值包含一個大的不同值的集合，且滿足 (1) 這個屬性沒有一般化的運算(這個屬性沒有定義概念階層)；或 (2) 它的較高層次概念可用其他屬性來定義；則這個屬性會從起始工作關係中移除
- 屬性一般化:如果起始工作關係中一個屬性的屬性值包含一個大的不同值的集合，而且這個屬性存在一組一般化的運算，則應選擇一個一般化的運算並對這個屬性進行套用
- 屬性一般化界限控制:屬性界限值為2到8，並應允許專家或使用者修改這個界限值
- 一般化關係界限控制: 控制最後關係/規則大小

屬性導向歸納: 基本方法

運算法則：**Attribute oriented induction**。對使用者資料探勘要求來探勘一個關聯式資料庫的一般化特徵。

輸入：一個 n 維度基礎長方體 B，也就是$(A_1, \cdots, A_n)$

- *DB* 爲一關聯式資料庫；
- *DMQuery* 爲一資料探勘查詢；
- *a_list* 爲一組屬性(包含屬性，a_i)；
- $Gen(a_i)$ 為在屬性 a_i 的一組概念階層或一般化運算元；
- *a_gen_thresh* (a_i)為對屬性 a_i 的屬性一般化界限值。

輸出：P 為一主要一般化關係。

方法：

1. $W \leftarrow$ get_task_relevant_data (*DMQuery, DB*); // 假設工作關係 W 包含工作相關資料。
2. prepare_for_generalization (W); // 執行如下。
   (a) 檢視 W 並對每個屬性 a_i 收集的不同值(請注意：如果 W 很大，可以透過對 W 進行取樣來完成)
   (b) 對每個屬性 a_i 決定它是否被移除。如果答案為否，則根據特定或預設屬性界限值來計算它的最小希望層次 L_i，並決定對應群組 (v, v')。其中 v 為在 W 中 a_i 的不同值，v' 為它在 L_i 中的相對應的一般化值。
3. $P \leftarrow$ generalization (W)
   在累計個數與計算其他聚合值時，主要一般化關係 P 可以透過將 W 中每個 v 值用它相對應 v' 值進行取代而產生。這個步驟可以用以下任一方法來有效地執行。
   (a) 對每一個一般化值組，利用二元搜尋法將值組插入一個排序過的主要關係 P：如果值組已經存在於 P 則增加相對應的個數與其他聚合值，否則則插入 P。
   (b) 由於大部分例子中在主要關係階層的不同值的個數是很小的，可以將主要關係編成一個 m 維陣列而 m 為 P 中的屬性個數，每個維度包含相對應的一般化屬性值。每個陣列元素儲存相對應個數與聚合值。一般化值組的插入可以透過相對應陣列元素的指標聚合來完成。

Example

- DMQL:描述Big University中研究生的一般特徵

```
use Big_University_DB
mine characteristics as "Science_Students"
in relevance to 姓名, 性別, 主修, 出生地, 生日, 居住地, 電話號碼, 平均成
    績 from 學生
where 學生 in "研究生"
```

- 對應的關聯式查詢:

```
use Big_University_DB
select 姓名, 性別, 主修, 出生地, 生日, 居住地, 電話號碼, 平均成績
from 學生
where 狀態 in { "M.Sc", "M.A.", "M.B.A.", "Ph.D." }
```

一般性的呈現

- 一般化關係:
  - 某些或全部屬性一般化關係, 包含各樹或其他聚合值
- 交互表格:
  - 將結果對應至交互表形式 (類似列聯表).
  - 圖式技巧:
  - 柱狀圖、圓形圖與曲線等.
- 數值特徵規則:
  - 一個一般化的值組比較不可能100%代表所有起始工作關係的值組。所以每個規則都有一個百分比，代表滿足規則左半部或右半部的資料值組的百分比, 例.,

$$\forall X, \text{項目}(X) = \text{“電腦”} \Rightarrow$$
$$(\text{位置}(X) = \text{“亞洲”})[t: 25.00\%] \vee (\text{位置}(X) = \text{“歐洲”})[t: 30.00\%] \vee$$
$$(\text{位置}(X) = \text{“北美”})[t: 45.00\%]$$

探勘類別比較

- 比較: 比較兩個或更多類別
- 方法:
  - 將資料根據目標類別與比較目標進行分割
  - 將兩個類別一般化至相同高層次概念
  - 利用相同高層次敘述來比較值組
  - 險是每個值組敘述與伴隨的兩個度量
    - 支持 – 單一類別內分佈
    - 比較 – 類別間分佈
  - 標註具有有效區別特性值組
- 相關分析:
  - 尋找最能區別不同類別屬性

數值性區別規則

- Cj =目標類別
- q_a =包含某些目標類別的值組
  - 有可能會包含一些對比類別的值組
- d-weight
  - 範維: [0, 1]

$$d-weight = \frac{count(q_a \in C_j)}{\sum_{i=1}^{m} count(q_a \in C_i)}$$

- 數值性區別規則形式

$$\forall X,\ target\_class(X) \Leftarrow condition(X)\quad [d : d\_weight]$$

範例:數值性區別規則

表4.20 一個一般化值組中研究生與大學生的個數分佈

狀態	主修	年齡範圍	平均成績	個數
graduate	Science	21...25	good	90
undergraduate	Science	21...25	good	210

- 數值性區別規則

$$\forall X, \text{狀態}(X) = \text{“研究生”} \Leftarrow \text{主修}(X) = \text{“科學”} \wedge \text{年齡範圍}(X) = \text{“21…25”} \wedge \text{平均成績}(X) = \text{“good”} \quad [d : 30\%] \tag{4.5}$$

  - 90/(90 + 210) = 30%

類別描述

- 數值性特徵規則

$$\forall X,\ target\_class(X) \Rightarrow condition(X)\ \ [t : t\_weight]$$

  - 必須

- 數值性區別規則

$$\forall X,\ target\_class(X) \Leftarrow condition(X)\ \ [d : d\_weight]$$

  - 足夠

- 數值性描述規則

$$\forall X,\ target\_class(X) \Leftrightarrow$$

$$condition_1(X)\,[\mathrm{t} : \mathrm{w}_1, \mathrm{d} : \mathrm{w}'_1] \vee \ldots \vee condition_n(X)\,[\mathrm{t} : \mathrm{w}_n, \mathrm{d} : \mathrm{w}'_n]$$

  - 必須而且足夠

範例:數值性描述規則

表4.22 同表4.21，但是對每個類別加上 t-weight 與 d-weight。

	項目								
	電視			電腦			兩個項目		
	項目	t-weight	d-weight	項目	t-weight	d-weight	項目	t-weight	d-weight
歐洲	80	25%	40%	240	75%	30%	320	100%	32%
北美	120	17.65%	60%	560	82.35%	70%	680	100%	68%
兩個區域	200	20%	100%	800	80%	100%	1000	100%	100%

- 對類別歐洲的數值性描述規則

$$\forall X, \text{住置}(X) = \text{“歐洲”} \Leftrightarrow$$
$$(\text{項目}(X) = \text{“電視”})[t:25\%, d:40\%]\theta$$
$$(\text{項目}(X) = \text{“電腦”})[t:75\%, d:30\%]$$

總結

- 有效率的資料方塊計算方法
  - Multiway陣列聚合
  - BUC
  - Star-cubing
  - 高維度OLAP
- 進一步資料方塊方法
  - 發掘導向探索
  - 多特性資料方塊
  - 方塊梯度分析
- 另外一般化方法:屬性導向歸納