Chapter 14

ASSOCIATION RULES (1)

(關聯規則)

背景

啤酒與尿布

背景

都市傳說 , 90 年代一名 Walmart 的主管在分析報表的時候發現

啤酒和尿布這二個沒有任何關係的物品

在特定的日期會出現在同一筆購物明細中 !!!

進一步的分析後 , 通常是年青爸爸會這樣一起買

Why !? ~

背景

在美國，一般是媽媽在家顧小孩，年輕的爸爸去超市買尿布

在購買尿布的同時，通常會順便買啤酒

賣場中若只能買到兩件商品之一

爸爸會直接到可以同時買到啤酒與尿布的超市

簡介

關聯規則探勘演算法首先在 1993 年被 IBM 所提出

目的就是要從大量的資料中找出有趣或相關的規則

而該研究中心隔年所提出的 Apriori 演算法到現今仍有大量的應用

簡介

PCHOME 24h (https://24h.pchome.com.tw/)

簡介

Amazon Japan (https://www.amazon.co.jp)

画像にマウスを合わせると拡大されます

この商品を買った人はこんな商品も買っています

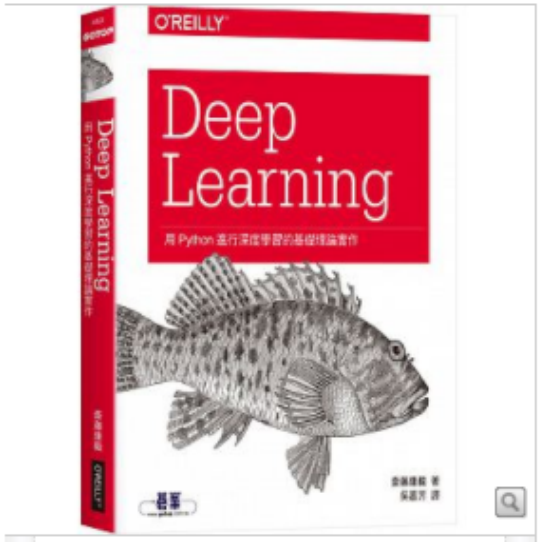
【PS4】スーパーロボット大戦X【早期購入特典】スーパーロボット大戦X「早期購入4大特典」プロダ…
バンダイナムコエン …
PlayStation 4
￥ 9,100

実況パワフルプロ野球2018（【初回限定特典】歴代パワプロシリーズオープニングテーマセットDLC…
コナミデジタルエン …
PlayStation Vita
￥ 6,163 prime

スーパーボンバーマンR（【初回限定特典】ボンバーマンキラキラ8兄弟セット&ゴールデンビックバ…
コナミデジタルエン …
PlayStation 4
￥ 3,509 prime

簡介

博客來 (http://www.books.com.tw/)



Deep Learning：用Python進行深度學習的基礎理論實作

作者：斎藤康毅 追蹤作者

譯者：吳嘉芳

出版社：歐萊禮 訂閱出版社新書快訊

出版日期：2017/08/17

語言：繁體中文

定價：~~580~~元

優惠價：79折 458 元

優惠期限：2018年03月31日止

抵用購物金最多再省$137 詳情

運送方式：

可配送點 :台灣、蘭嶼、綠島、澎湖、金門、馬祖、全球

可取貨點 :台灣、蘭嶼、綠島、澎湖、金門、馬祖

買了此商品的人，也買了...

精通 Python：運用簡單的套件進行現代運算

原價：780

優惠價：79折616元

TensorFlow+Keras深度學習人工智慧實務應

原價：590

優惠價：79折466元

簡介

項目集 (itemset)

在一個購物籃中一件商品就代表一個 item

所多商品 (k) 就購成一個 k-itemset

TID	Transaction
10	a, b, c, d, f
20	b, c, d, f, g, h
30	a,c, d, e, f
40	c, e, f, g

number	itemset
1	a
2	b, c
3	a, c, d

簡介

關聯規則 (Association Rule)

給定一條 Rule : X->Y, X 稱為先決條件，而 Y 稱為對應的關聯

舉例來說 啤酒 -> 尿布代表的就是買啤酒的人就會一起買尿布

而一條 Rule 是否具有可性度，基本上是由以下二個指標來衡量

支持度 (Support)

信心度 (Confidence)

簡介

支持度 (support)

Fraction of the transactions that contain both *A* and *B*

(指的該項目集是同時出現 A 和 B 的機率)

TID	Transaction
10	a, b, c, d, f
20	b, c, d, f, g, h
30	a,c, d, e, f
40	c, e, f, g

itemset	support
a	3/4 =75%
ac	2/4 = 50%
acd	2/4 = 50%

簡介

最小支持度 (minimum support, min_sup)

Min_sup 是一個門檻值 (threshold), 用來將支持度較低的 itemset 過濾掉

(而留下來的項目集係為頻繁項目集 , frequent pattern, frequent itemset)

TID	Transaction	Min_sup
10	a, b, c, d, f	**60%**
20	b, c, d, f, g, h	
30	a,c, d, e, f	
40	c, e, f, g	

itemset	support	Frequent?
a	3/4 =75%	**Yes**
ac	2/4 = 50%	**No**
acd	2/4 = 50%	**No**

簡介

信心度 (confidence)

Fraction of the transactions contain A also contain B

(A 先發生的情況下 B 發生的機率)

TID	Transaction
10	a, b, c, d, f
20	b, c, d, f, g, h
30	a,c, d, e, f
40	c, e, f, g

Rule	Support	Confidence
a->b	a(2), ab(1)	1/2 = 50%
c->a	c(4), ac(2)	2/4 = 50%
c->d	c(4), cd(3)	3/4 = 75%

簡介

最小信心度 (minimum confidence, min_conf)

Min_conf 也是一個門檻值 (threshold), 用來將信心度較低的 Rule 過濾掉

(而留下來的規則為強關聯規則 , Strongly Rules)

TID	Transaction	Min_conf
10	a, b, c, d, f	**60%**
20	b, c, d, f, g, h	
30	a, c, d, e, f	
40	c, e, f, g	

Rule	Confidence	Strongly?
a->b	1/2 = 50%	No
c->a	2/4 = 50%	No
c->d	3/4 = 75%	Yes

簡介

提升度 (lift)

Fraction of the transactions that contain both Rule(A->B) and B

(A->B rule 的機率 /B 發生的機率，越大越好)

TID	Transaction
10	a, b, c, d, f
20	b, c, d, f, g, h
30	a,c, d, e, f
40	c, e, f, g

Rule	Confidence	lift
a->b	1/2 = 50%	50%/25% = 2
c->a	2/4 = 50%	50%/50% = 1

簡介

關聯規則探勘演算法 (Association Rule Mining Algorithm)

Step 1: 選出支持度大於最小支持度的項目集，也就是頻繁項目集 (FP)

Step 2: 從 FP 當中找出滿足最小信心度的所有規則

Note : Minsup 通常不會太大 (ex 5-10%)

Minsup 通常不會太小 (ex 50-75%)

Apriori Algorithm

Apriori 利用逐層搜尋的疊代法 (Level-Wise Search) 來找出頻繁項目集

再以最小信心度為條件計算出所有的關聯規則

Apriori 有個重要的特性：反單調性 (anti-monotonicity)

如果一個子集合無法滿足最小支持度

則它所有的超集合 (super set) 也不會滿足最小支持度

Apriori Algorithm

Apriori 完整演算法

Step 1: 掃瞄第一次交易資料庫找出頻繁項目集 L_1

Step 2: 利用結合法 (join step) 將兩兩一組的 L_k 產生新的候選項目集 C_{k+1}, (k>= 1)

Step 3: 利用反單調性將 C_{k+1} 進行第一次候選項目集修剪 (prune step)

Step 4: 計算每一個候選項目集 C_{k+1} 的支持度

Step 5: 將 C_{k+1} 進行第二次修剪，留下滿足最小支持度的頻繁項目集 L_k

Step 6: 重覆 Step 2 to Step 4 直到無任何頻繁項目集產生

Apriori Algorithm

join Step

利用結合法 (join step) 將兩兩一組的 L_k 產生新的候選項目集 C_{k+1}

Case 1. L_1 的所有項目集可以直接兩兩結合成 C_2

Case 2. (k>1) 若任二個 L_k 前 $k-1$ 項都一樣，最後一項不同，則可結合產生 C_{k+1}

L_1	C_2	
A	AB	BD
B	AC	CD
C	AD	
D	BC	

L_2	C_3
AB	ABC
AC	BCE
BC	
BE	

Apriori Algorithm

min_sup = 2

TID	Transaction
1	A, C, D
2	A, B, C
3	A, C
4	B, E
5	B,C,E

Scan TDB to get support of C_1 *and* L_1

itemset	support	Frequent?
A	3	Yes
B	3	Yes
C	4	Yes
D	1	No
E	2	Yes

itemset	support
A	3
B	3
C	4
E	2

Apriori Algorithm

min_sup = 2

To generate C_2 *from* L_1

itemset	support
A	3
B	3
C	4
E	2

itemset	support
AB	1
AC	3
AE	0
BC	2
BE	2
CE	1

To get L_2 *from* C_1

itemset	support
AC	3
BC	2
BE	2

To generate C_3 *from* L_2

Anti-monotone

itemset	support
BCE	1

TID	Transaction
1	A, C, D
2	A, B, C
3	A, C
4	B, E
5	B,C,E

Apriori Algorithm

min_sup = 2

TID	Transaction
1	A, C, D
2	A, B, C
3	A, C
4	B, E
5	B,C,E

All Frequent Pattern

L_1	L_2
A	AC
B	BC
C	BE
E	

Apriori Algorithm

min_conf = 60%

TID	Transaction
1	A, C, D
2	A, B, C
3	A, C
4	B, E
5	B,C,E

To generate Rules

L_2	Rule	Confidence
AC	A->C C->A	(AC)/A=2/3=66% (AC)/C=2/4=50%
BC	B->C C->B	(BC)/B=2/3=66% (BC)/C=2/4=50%
BE	B->E E->B	(BE)/B=2/3=66% (BE)/E=2/2=100%

Rules	
A->C	C->A
B->C	B->E
E->B	

Frequent Pattern-Growth Algorithm

FP-growth 演算法則不需要產生候選項目集

而是將資料集壓縮成頻繁樣式樹的格式 (frequent-pattern tree)

接著再接壓縮過後的資料集切割成數個條件資料庫 (conditional databases)

每一個條件資料庫對應到一個頻繁項目或樣式片段 (pattern fragment)

Frequent Pattern-Growth Algorithm

FP-Growth 完整演算法

Step 1: 掃瞄第一次交易資料庫找出頻繁項目集 L_1, 並由支持度大到小儲存

Step 2: 掃瞄第二次交易資料庫依照 L_1 順序來將所有項目建立分支 (branch)

Step 3: 從每一個 L_1 開始建立條件樣式集與條件 FP-tree (conditional FP-tree)

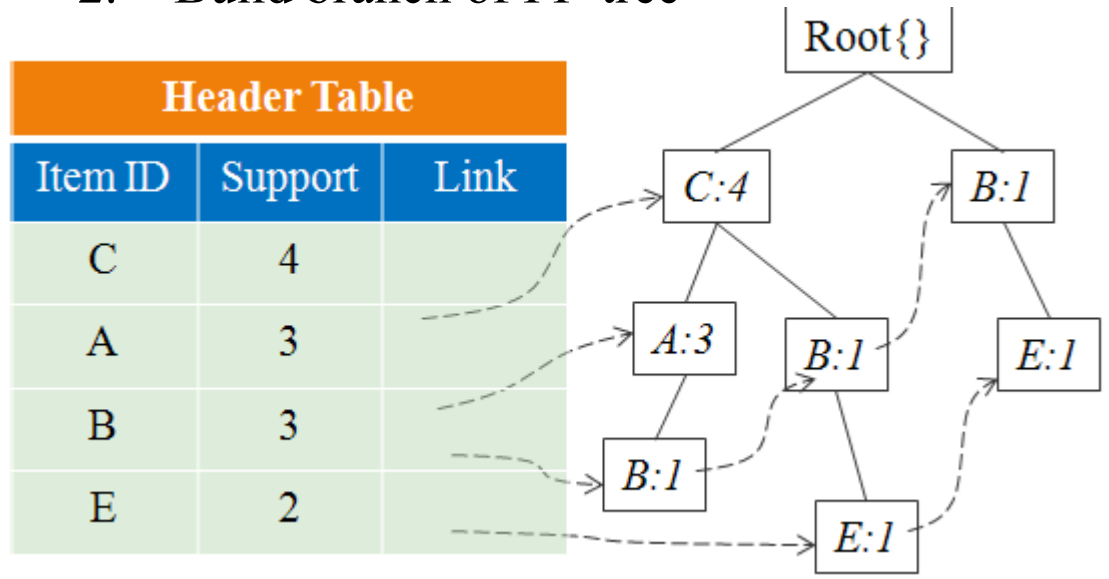
Step 4: 遞迴此 FP-tree 直到條件 FP 樹僅包含單一路徑為止

FP- Growth Algorithm min_sup = 2

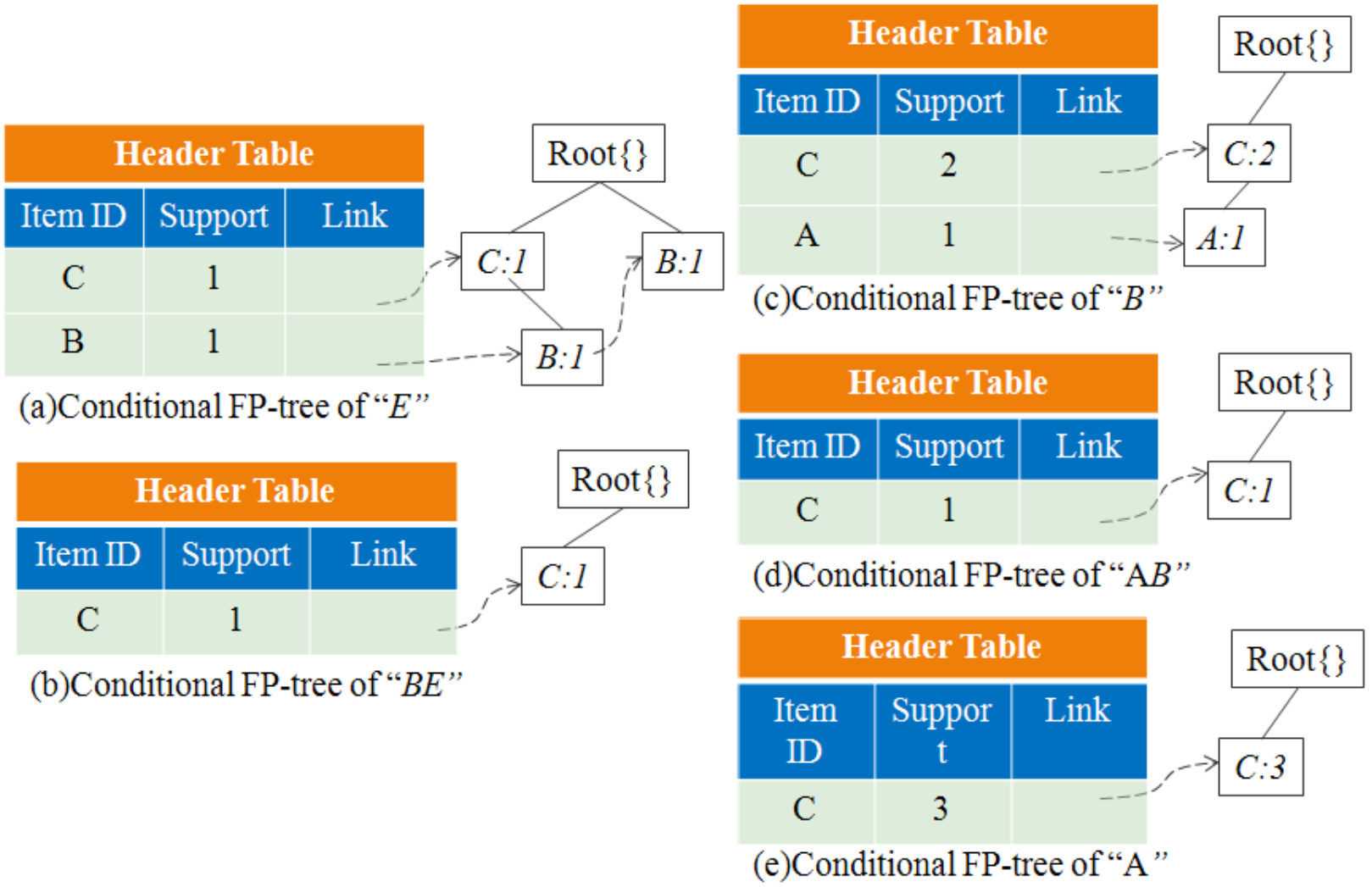
TID	Transaction
1	A, C, D
2	A, B, C
3	A, C
4	B, E
5	B,C,E

1. Scan TDB to get support of L_1
2. Build branch of FP-tree

Header Table		
Item ID	Support	Link
C	4	
A	3	
B	3	
E	2	



FP- Growth Algorithm min_sup = 2



ECLAT Algorithm

ECLAT 演算法是將 TID 為索引的資料庫轉成以 item 為索引的資料庫

也就是說 item 會變成 鍵 (key), 而資料庫中的每一個 item 會找到對應的 TID

ECLAT 的好處就是對於資料庫的掃瞄只會有一次

接下來只要進行集合交集的動作就可以算出所有的 support

ECLAT Algorithm

min_sup = 2

TID	Transaction
1	A, C, D
2	A, B, C
3	A, C
4	B, E
5	B,C,E

Horizontal database to Vertical

A	B	C	D	E
1 2 3	2 4 5	1 2 3 5	1	4 5

Support counting to get L_1

itemset	support
A	3
B	3
C	4
E	2

ECLAT Algorithm

min_sup = 2

Horizontal database to Vertical

A	B	C	D	E
1 2 3	2 4 5	1 2 3 5	1	4 5

itemset	Support vector
A	11100
B	01011
C	11101
E	00011

itemset	Support Counting
AB (A and B)	01000
AC (A and C)	11100
BC (B and C)	01001
CE (C and E)	00001

隨堂練習

1. 給定一個交易庫如下表所示 :

TID	Transaction
1	1, 2, 5
2	2, 4
3	2, 3
4	1, 2, 4
5	1, 3
6	2, 3
7	1, 3
8	1, 2, 3, 5
9	1, 2, 3

2. 利用 Apriori 演算找出 frequent pattern (假設 min_sup =2)
3. 找出所有的關聯規則

Any Questions !?