Chapter 13 – Association Rules

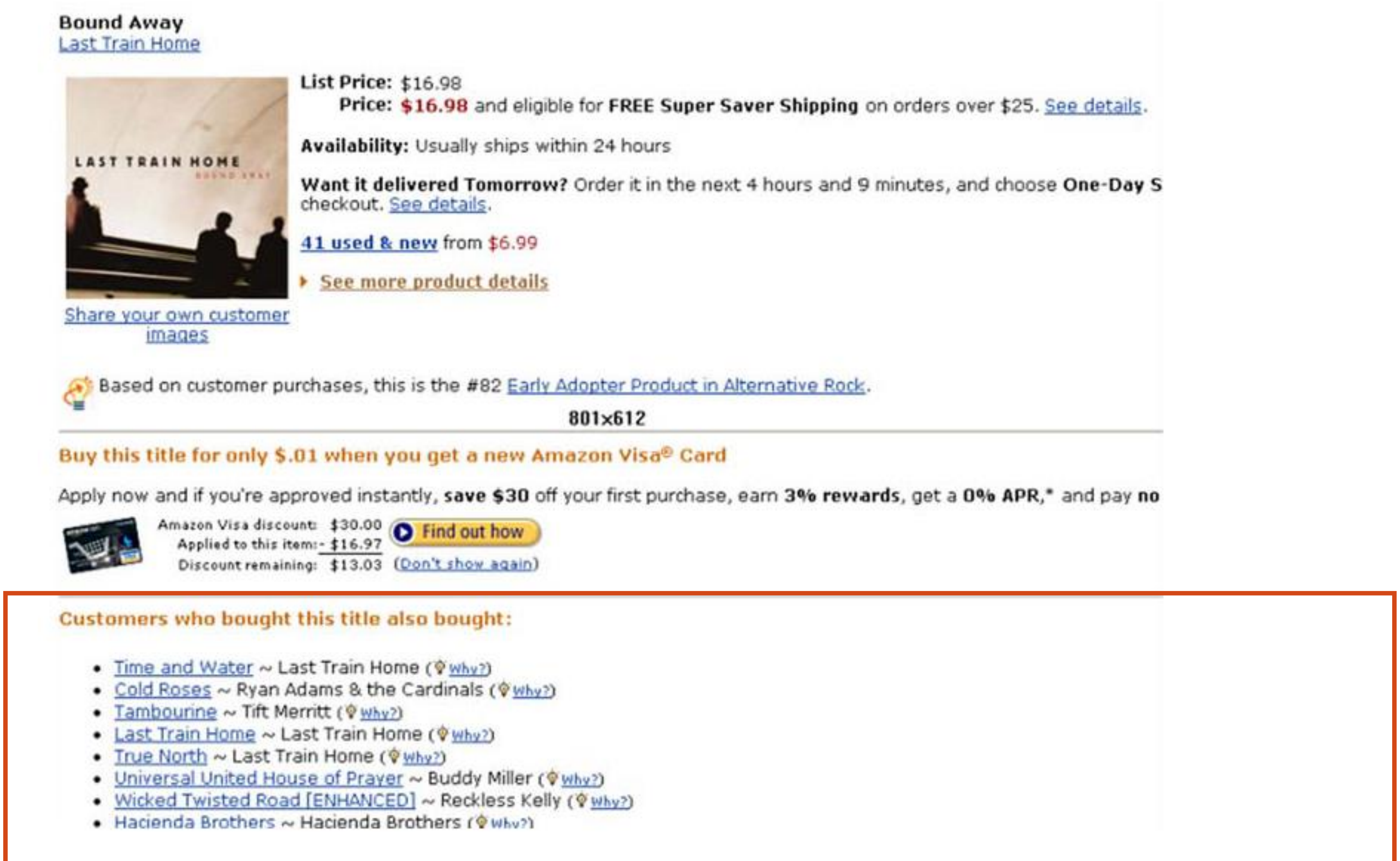
Data Mining for Business Intelligence

Shmueli, Patel & Bruce

What are Association Rules?

- Study of “what goes with what” co-occurrence = 동시 발생
  - “Customers who bought X also bought Y”
  - What symptoms go with what diagnosis
- Transaction-based or event-based
- Also called “market basket analysis” and “affinity analysis”
- Originated with study of customer transactions databases to determine associations among items purchased

Used in many recommender systems



Generating Rules

Terms

"IF" part = **antecedent**

"THEN" part = **consequent**

"Item set" = the items (e.g., products) comprising the antecedent or consequent

- Antecedent and consequent are *disjoint* (i.e., have no items in common)

Tiny Example: Phone Faceplates

Transaction	Faceplate Colors Purchased			
1	red	white	green	
2	white	orange		
3	white	blue		
4	red	white	orange	
5	red	blue		
6	white	blue		
7	white	orange		
8	red	white	blue	green
9	red	white	blue	
10	yellow			

Many Rules are Possible

For example: Transaction 1 supports several rules, such as

- “If red, then white” (“If a red faceplate is purchased, then so is a white one”)
- “If white, then red”
- “If red and white, then green”
- + several more

Frequent Item Sets

- Ideally, we want to create all possible combinations of items
- **Problem:** computation time grows exponentially as # items increases
- **Solution:** consider only “frequent item sets”
- Criterion for frequent: *support*

Support

Support = # (or percent) of transactions that include both the antecedent and the consequent

Example: support for the item set {red, white} is 4 out of 10 transactions, or 40%

Apriori Algorithm
“선험” 알고리즘

Apriori property

- “실제 계산하지 않아도 (경험하지 않아도) 선험적으로 안다”
- If set X is frequent, any of its “nonempty” subset is frequent
- {빵,우유} 포함하는 transaction 비율이 5% 이상이면, {빵}, {우유}를 포함하는 transaction 비율이 모두 5% 이상임

Apriori property

- 대우
- If set X is NOT frequent, any **superset** containing X is NOT frequent

- If X = {D} is NOT frequent, {A, D} is NOT frequent.
- If X = {A, B} is NOT frequent, {A, B, C} is NOT frequent.
- {빵, 우유} 포함하는 transaction 비율이 5% 미만이면, {빵, 우유, 버터} 포함하는 transaction 비율도 5% 미만

Apriori property

- 알고리즘은 1 item set 을 먼저 구성한 후, 2 item set, 3 item set 순으로 차차 늘여 감
- 각 단계에서는 generate & scan 를 반복
  - Generate : 후보 itemset 구축
  - Scan : 실제 support 값을 계산하여 걸러냄
    - 전체 database 를 scan! 계산량
- a priori property 의 효과는?

Generating Frequent Item Sets

For *k* products…

1. User sets a minimum support criterion
2. Next, generate list of one-item sets that meet the support criterion
3. Use the list of one-item sets to generate list of two-item sets that meet the support criterion
4. Use list of two-item sets to generate list of three-item sets
5. Continue up through *k*-item sets

Generating Frequent Item Sets

For *k* products…

1. User sets a minimum support criterion
2. Next, generate list of one-item sets that meet the support criterion = **generate & scan**
3. Use the list of one-item sets to generate list of two-item sets that meet the support criterion
4. Use list of two-item sets to generate list of three-item sets
5. Continue up through *k*-item sets

The Apriori Algorithm—An Example

$Sup_{min} = 2$

Database TDB

Tid	Items
10	A, C, D
20	B, C, E
30	A, B, C, E
40	B, E

1^{st} scan

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}

2^{nd} scan

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}

3^{rd} scan

L_3

Itemset	sup
{B, C, E}	2

The Apriori Algorithm—An Example

- L2 에서 C3 도출 과정

Itemset	sup
{A, C}	2
{B, C}	2
{B, E}	3
{C, E}	2

Itemset
{B, C, E}

- 후보 3-itemset: A,B,C / A,B,E / B,C,E frequent 한가?

Measures of Performance

- ***Confidence*:** the % of antecedent transactions that also have the consequent item set ~= 조건부 확률
- Confidence (A=>B) ~= P(B|A)

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

Tid	Items
10	A, C, D
20	B, C, E
30	A, B, C, E
40	B, E

- A => B / B => A / ... 의 confidence 값은?

Measures of Performance

- ***Confidence*:** the % of antecedent transactions that also have the consequent item set ~= 조건부 확률

Itemset
{B, C, E}

Tid	Items
10	A, C, D
20	B, C, E
30	A, B, C, E
40	B, E

- B => C, E / C => B, E / E => B, C 값은?
- B, C => E / B, E => C / C, E => B 값은?

Measures of Performance

- **Lift** = *confidence*/(*benchmark confidence*)
- *Benchmark confidence* = transactions with consequent as % of all transactions
- Lift (A=>B) = Confidence (A=>B) / Confidence (B)

  ~= P(B|A) / P(B)
- Lift > 1 indicates a rule that is useful in finding consequent items sets (i.e., more useful than just selecting transactions randomly)

Measures of Performance

- **Lift** = *confidence*/(*benchmark confidence*)
- B 가 원래 frequent 한 것은 아닌지…
- Confidence (콜라 => 햄버거) high
- Confidence (다이어트콜라 => 햄버거) high 라도
- 그러나 P(햄버거) 가 높다면, 별 의미 없음

Measures of Performance

- **Lift** = *confidence*/(*benchmark confidence*)

Tid	Items
10	A, C, D
20	B, C, E
30	A, B, C, E
40	B, E

- A => B / B => A 의 lift는
- B => C, E / C => B, E / E => B, C 값은?
- B, C => E / B, E => C / C, E => B 값은?

Alternate Data Format: Binary Matrix

Transaction	Red	White	Blue	Orange	Green	Yellow
1	1	1	0	0	1	0
2	0	1	0	1	0	0
3	0	1	1	0	0	0
4	1	1	0	1	0	0
5	1	0	1	0	0	0
6	0	1	1	0	0	0
7	1	0	1	0	0	0
8	1	1	1	0	1	0
9	1	1	1	0	0	0
10	0	0	0	0	0	1

Process of Rule Selection

Generate all rules that meet specified support & confidence

- Find frequent item sets (those with sufficient support – see above)
- From these item sets, generate rules with sufficient confidence

Example: Rules from {red, white, green}

{red, white} => {green} with confidence = 2/4 = 50%

- [(support {red, white, green})/(support {red, white})]

{red, green} => {white} with confidence = 2/2 = 100%

- [(support {red, white, green})/(support {red, green})]

Plus 4 more with confidence of 100%, 33%, 29% & 100%

If confidence criterion is 70%, report only rules 2, 3 and 6

All Rules (XLMiner Output)

Rule #	Conf. %	Antecedent (a)	Consequent (c)	Support(a)	Support(c)	Support(a U c)	Lift Ratio
1	100	Green=>	Red, White	2	4	2	2.5
2	100	Green=>	Red	2	6	2	1.666667
3	100	Green, White=>	Red	2	6	2	1.666667
4	100	Green=>	White	2	7	2	1.428571
5	100	Green, Red=>	White	2	7	2	1.428571
6	100	Orange=>	White	2	7	2	1.428571

Interpretation

- *Lift ratio* shows how effective the rule is in finding consequents (useful if finding particular consequents is important)
- *Confidence* shows the rate at which consequents will be found (useful in learning costs of promotion)
- *Support* measures overall impact

Caution: The Role of Chance

- Random data can generate apparently interesting association rules
- The more rules you produce, the greater this danger
- Rules based on large numbers of records are less subject to this danger
- 대응책은 human inspection

Example: Charles Book Club

ChildBks	YouthBks	CookBks	DoItYBks	RefBks	ArtBks	GeogBks	ItalCook	ItalAtlas	ItalArt	Florence
0	1	0	1	0	0	1	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	1	0	1	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0
1	0	0	0	0	1	0	0	0	0	1
0	1	0	0	0	0	0	0	0	0	0
0	1	0	0	1	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	0
1	1	1	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0

Row 1, e.g., is a transaction in which books were bought in the following categories: Youth, Do it Yourself, Geography

XLMiner Output

Rule #	Conf. %	Antecedent (a)	Consequent (c)	Support(a)	Support(c)	Support(a U c)	Lift Ratio
1	100	ItalCook=>	CookBks	227	862	227	2.320186
2	62.77	ArtBks, ChildBks=>	GeogBks	325	552	204	2.274247
3	54.13	CookBks, DoItYBks=>	ArtBks	375	482	203	2.246196
4	61.98	ArtBks, CookBks=>	GeogBks	334	552	207	2.245509
5	53.77	CookBks, GeogBks=>	ArtBks	385	482	207	2.230964
6	57.11	RefBks=>	ChildBks, CookBks	429	512	245	2.230842
7	52.31	ChildBks, GeogBks=>	ArtBks	390	482	204	2.170444
8	60.78	ArtBks, CookBks=>	DoItYBks	334	564	203	2.155264
9	58.4	ChildBks, CookBks=>	GeogBks	512	552	299	2.115885
10	54.17	GeogBks=>	ChildBks, CookBks	552	512	299	2.115885
11	57.87	CookBks, DoItYBks=>	GeogBks	375	552	217	2.096618
12	56.79	ChildBks, DoItYBks=>	GeogBks	368	552	209	2.057735

- Rules arrayed in order of lift
- Information can be compressed
  - e.g., rules 2 and 7 have same trio of books

Practical Tips

- Item 선택: Proper Level of Abstraction 입도
  - Beverage: Cola : Coca Cola : Coca Cola 250ml: Diet Coca Cola 250ml

Practical Tips

- Virtual Items
  - Day of the week, Time of the day, Season, Region, Shopper's Gender/age
    - "IF shopper in 20's AND night THEN cup ramen"
    - 지역적, 시간적, 계절적 차이 파악 가능
    - 지점 별 차이점도
  - Membership ID 정보
    - Other behavior 와의 연결을 통한 이해
    - 고객 주소, 과거 구매 내역
    - 포인트 카드

Practical Tips

- Other than Shopping basket?

- Transaction 으로 볼 수 있는 모든 items
  - 일련의 공정/검사를 거치는 경우, 각각이 item

  - 일련의 customer 구매 내역

Summary

- Association rules (or *affinity analysis*, or *market basket analysis*) produce rules on associations between items from a database of transactions
- Widely used in **recommender systems**
- Most popular method is **Apriori algorithm**
- To reduce computation, we consider only “frequent” item sets (=support)
- Performance is measured by *confidence* and *lift*
- Can produce a profusion of rules; review is required to identify useful rules and to reduce redundancy