Association Rules – the Apriori Algorithm

V.S. Subrahmanian
University of Maryland
vs@cs.umd.edu

Copyright (C) 2014, V.S. Subrahmanian

Basic Idea

- An association rule (AR) tries to find dependencies of the form $A_1, .., A_n \rightarrow B$ where $A_1, .., A_n, B$ are attributes (or conditions over attributes). *B* is different from all of the *A*'s.
- Intuitively this says: When $A_1, .., A_n$ occur together, *B* is also likely to occur.
- The *"goodness"* of an AR is captured by two quantities
  – Support,
  – Confidence

Copyright (C) 2014, V.S. Subrahmanian

Classical Example: Market Basket Analysis

- You work for a grocery store.
- Every time a person checks out, the system identifies the set of items the person bought.
- Thus, each transaction at the checkout register is a row in a table and the items bought are listed.

Transaction	Items
1	A,B,D
2	A,B,F
3	B,C,F
4	C,E
5	A,C,F
6	A,B,E
7	B,E,F

6 items: A, B, C, D, E, F

Copyright (C) 2014, V.S. Subrahmanian

Itemset

- An *itemset* is a set of items – in our example, any subset of {A,B,C,D,E,F}.
- An itemset is *frequent* if it occurs often enough, i.e. if it occurs over a certain number of times.
- Suppose *r* = $A_1, . . , A_n \rightarrow B$ is an AR.
- Support(r) = Probability that a random transaction contains $\{A_1, \ldots, A_n, B\}$. NOTE: *Sometimes people just use the number of transactions whose itemsets contain* $\{A_1, \ldots, A_n, B\}$.
- Confidence(r) = Probability of *B* being in an itemset, given that $\{A_1, \ldots, A_n\}$ are in it.

Copyright (C) 2014, V.S. Subrahmanian

Classical Example: Market Basket Analysis

- Let r = $A \rightarrow B$
- Support(r) = 3/7 as A,B occur together in 3 out of 7 transactions.
- Confidence(r) = ¾ as B occurs in 3 out of 4 transactions in which A occurs.

Transaction	Items
1	A,B,D
2	A,B,F
3	B,C,F
4	C,E
5	A,C,F
6	A,B,E
7	B,E,F

6 items: A, B, C, D, E, F

Copyright (C) 2014, V.S. Subrahmanian

Classical Example: Market Basket Analysis

- Let r = $A, B \rightarrow F$
- Support(r) = 1/7 as A,B,F occur together in 1 out of 7 transactions.
- Confidence(r) = 1/3 as F occurs in 1 out of 3 transactions in which A and B both occur.

Transaction	Items
1	A,B,D
2	A,B,F
3	B,C,F
4	C,E
5	A,C,F
6	A,B,E
7	B,E,F

6 items: A, B, C, D, E, F

Copyright (C) 2014, V.S. Subrahmanian

Association Rule Mining Problem

- Given a database DB having the schema (Transaction,Itemset) and to integers *s,c* find all association rules *r* having
  - $Support(r) \geq s$,
  - $Confidence(r) \geq c$.

Copyright (C) 2014, V.S. Subrahmanian

Apriori Algorithm

- For each $i \geq 1, L_i$ denotes the set of all frequent itemsets of cardinality *i.*
- The idea is to iteratively expand L_i to $L_{\{i+1\}}$.
- Once $L_j = \emptyset$ for some *j,* we can stop.
- At this stage. $L_1, L_2, \ldots, L_{\{j-1\}}$ would represent the set of all frequent itemsets (i.e. satisfying the support requirement).
- Check these to see if the confidence levels hold.

Copyright (C) 2014, V.S. Subrahmanian

Join Operation

- Compute $L_{\{j+1\}}$ by joining L_j with itself.
- Suppose *j=2* and $L_j = \{\{A, B\}, \{A, C\}, \{C, D\}\}$.
- The *join* of L_j with itself is
  - {A,B,C}: join {A,B} with {A,C}
  - {A,B,C,D}: join {A,B} with {C,D} but rejected in join as it has 4 elements;
  - {A,C,D}: join {A,C} with {C,D}.
- So the returned join is {{A,B,C},{A,C,D}}.

Copyright (C) 2014, V.S. Subrahmanian

Pruning Step

- **Theorem.** If *X* is a frequent itemset (i.e. it occurs in a sufficiently high percentage of transactions) then so must any subset $Y \subseteq X$.
- **Proof ?**
- **Implication.**
  - If *X* is a candidate to be inserted into L_i but some subset *Y* of cardinality (i-1) is not in $L_{\{i-1\}}$, then *X* should not go into L_i.

Copyright (C) 2014, V.S. Subrahmanian

Apriori Algorithm, Phase I: Find AR conditions having enough support

$L_1 = \{ \{x\} \mid x \ is \ an \ item\}$; %singletons

j=1;

While $(L_j \neq \emptyset)$ $\boldsymbol{do}$

{ j=j+1;

$C_j = join(L_j, L_j)$; % find candidates

$L_j = \{x \mid x \in C_j \& \ support(x) \geq c\}$;

}

Return $\bigcup_j L_j$

Copyright (C) 2014, V.S. Subrahmanian

Classical Example: Market Basket Analysis

- Let *s = 0.2.*
- COUNTs are as follows:

Transaction	Items
1	A,B,D
2	A,B,F
3	B,C,F
4	C,E
5	A,C,F
6	A,B,E
7	B,E,F

Item	COUNT
A	4
B	5
C	3
D	1
E	3
F	4

So $L_1 = \{\{A\},\{B\},\{C\},\{E\},\{F\}\}$

Copyright (C) 2014, V.S. Subrahmanian

Classical Example: Market Basket Analysis

- Let *s = 0.2.*
- *Pruning Step:* Nothing containing D can be in L_2. *Why?*
- Instead of considering 18 pairs, we only need to consider10 pairs.
- What makes it into L_2?
- {A,B}:3,{A,F}:2
- {B,E}:2,{B,F}:2
- {C,F}:2
- All of these occur at least twice in the data, so support is enough.

Transaction	Items
1	A,B,D
2	A,B,F
3	B,C,F
4	C,E
5	A,C,F
6	A,B,E
7	B,E,F

So
$L_2 = \{\{A,B\},\{A,F\},\{B,E\},\{B,F\},\{C,F\}\}$

Copyright (C) 2014, V.S. Subrahmanian

Classical Example: Market Basket Analysis

- What makes it into L_3?
- {A,B,F}:1
- {A,B,E}: 1. Can prune because AE is not in L_2.
- {A,C,F}:1. Can prune because AC is not in L_2.
- {A,B,E,F}:0 X
- {B,E,F}:1. Can prine because EF is not in L_2.
- {B,C,F}:1. Can prune as BC is not in L_2.

Transaction	Items
1	A,B,D
2	A,B,F
3	B,C,F
4	C,E
5	A,C,F
6	A,B,E
7	B,E,F

So $L_3 = \emptyset$.
So our iteration can stop.

Copyright (C) 2014, V.S. Subrahmanian

Apriori Algorithm, Phase II: Find ARs satisfying confidence condition

SOL = {};

foreach itemset X, |X|$\geq$2 ret. by Phase I **do**

  foreach a in X **do**

    if $conf(X - \{a\} \rightarrow a) \geq c$ then

      SOL = SOL **U** $\{X - \{a\} \rightarrow a\}$;

Return SOL.

Copyright (C) 2014, V.S. Subrahmanian

Apriori Algorithm, Phase II: Find ARs satisfying confidence condition

SOL = {};

foreach itemset X, |X|≥2 ret. by Phase I **do**

foreach $a \subseteq X\ s.t. |X - a| \geq 1$ **do**

if $conf(X - \{a\} \rightarrow a) \geq c$ then

SOL = SOL **U** $\{X - \{a\} \rightarrow a\}$;

Return SOL.

Allows rule heads to have multiple items.

Copyright (C) 2014, V.S. Subrahmanian

Candidate Itemsets

- Itemsets with enough support are:
- $L = \left\{ \begin{array}{c} \{A\},\{B\},\{C\},\{E\}, \\ \{F\},\{A,B\},\{A,F\}, \\ \{B,E\},\{B,F\},\{C,F\} \end{array} \right\}.$
- Of these, the only ones that can give rise to a rule are the doubletons [singletons can't generate a rule – why?]

Assoc. Rule	Confidence
A => B	
B => A	
A => F	
F => A	
B => E	
E => B	
B => F	
F => B	
C => F	
F => C	

Candidate Itemsets

Transaction	Items
1	A,B,D
2	A,B,F
3	B,C,F
4	C,E
5	A,C,F
6	A,B,E
7	B,E,F

Assoc. Rule	Confidence
A => B	¾ = 75%
B => A	3/5 = 60%
A => F	¼ = 25%
F => A	2/4 = 50%
B => E	2/5 = 50%
E => B	2/3 = 66.67%
B => F	3/5 = 60%
F => B	¾ = 75%
C => F	2/3 = 66.67%
F => C	2/4 = 50%

The association rules discovered are now based on the confidence threshold. For example, if the threshold is 65%, then the rules returned are highlighted in red.

Copyright (C) 2014, V.S. Subrahmanian

Are ARs good?

- Not necessarily – why?
- Give an example

Copyright (C) 2014, V.S. Subrahmanian

Lift

- Lift($A \to B$) = $\frac{\boldsymbol{P}(B|A)}{\boldsymbol{P}(B)}$.
- Intuition:
  – Rule could have high support.
  – Rule could have high confidence.
  – But if **P**(B) is almost the same as **P**(B|A)**,** then A could not have much to do with B being true.
  – On the other hand, if “lift” is high, then having A be true makes a difference in whether B is true.

Copyright (C) 2014, V.S. Subrahmanian

Flaws with A Priori Algorithm

- Too slow !
- Number of possible candidates (C_j step) can be enormous when the join is done in Phase I.

Copyright (C) 2014, V.S. Subrahmanian

In Class Exercise

Transaction	Items
1	A,B,D
2	A,B,E,F
3	A,B,C,F
4	C,E,D
5	A,C,E,F
6	A,B,E
7	B,C,F
8	A,D,E,F
9	A,C,D
10	B,E,F

Support = 3/10
Confidence = 55%

Copyright (C) 2014, V.S. Subrahmanian