Product Recommendation using Error-Tolerant Association Rules (ETAR)

Anthony K. H. Tung
Intelligent Database Laboratory

Outline

- Introduction
- Motivation
  - Why association rules ?
  - Why error-tolerant ?
- Discovering ETAR
- Using ETAR
- Conclusion

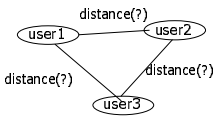
Introduction

- Problem Statement
  - Given a set of users and the products they brought or rank, try to recommend other products which they will be interested in buying.
- Running example:
  - An online bookstore like Amazon.com

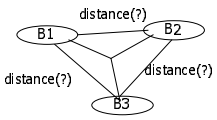
Why Association Rules (I)?

- Association rules represent a different philosophy from other collaborative filtering algorithm

Row Clustering



Column Clustering



Why Association Rules (II)?

- Collaborative filtering algorithms like Grouplens and Ringo are row clustering algorithms which judge the similarity between two users by considering their rating on **ALL** products (which could be say 10 000).
- Potential problem

Users	Computer Related			Ice Hockey			Snooker			Ballet	
	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11
User1	5	5	-	5	5	5	1	1	1	-	-
User2	5	5	5	1	1	1	5	5	5	3	3
User3	3	3	3	5	5	5	1	1	1	5	5

Why Association Rules (III)?

- Association rules focus on finding items which are often bought together.
- Thus, it is able to consider a subset of the items as oppose to comparing the rating on all items between two users.

Why ETAR (I)?

- An example of error tolerant association rules is of the form
  - R_1:(Any 3 of {a,b,c,d}) -> d
    - (supp:20, conf:80%)
- supp(R_1)= |U|,
  - U=set of users that match at least 3 of {a,b,c,d}.
- conf(R_1)= % of users in U that buy item d.
- Instead of needing to match all of LHS, only partial matching is needed in order to trigger the rule.

Why ETAR (II)?

- Main considerations:
  - Frequent pattern take time to growth.
  - Catch 22 situation :
    - Without the proper recommendation, patterns may not become frequent.
    - When patterns are not frequent, we can't make proper recommendation.
- ETAR solve the problem partially by consider "immature" frequent pattern.

Why ETAR (III)?

- How ETAR came about ?

	B1	B2	B3	B4	B5
U1	-	1	-	-	-
U2	-	-	1	-	-
U3	1	-	-	-	-
U4	-	-	-	1	-
U5	-	-	-	-	1

Month 1

	B1	B2	B3	B4	B5
U1	1	1	-	-	-
U2	-	-	1	1	-
U3	1	1	-	-	-
U4	-	1	-	1	-
U5	-	-	1	-	1

Month 2

	B1	B2	B3	B4	B5
U1	1	1	-	1	-
U2	1	-	1	1	-
U3	1	1	-	-	1
U4	1	1	-	1	-
U5	-	1	1	-	1

Month 3

	B1	B2	B3	B4	B5
U1	1	1	-	1	1
U2	1	1	1	1	-
U3	1	1	-	1	1
U4	1	1	1	1	-
U5	1	1	1	-	1

Month 4

	B1	B2	B3	B4	B5
U1	1	1	1	1	1
U2	1	1	1	1	1
U3	1	1	1	1	1
U4	1	1	1	1	1
U5	1	1	1	1	1

Month 5

Desirable Property of ETAR

	B1	B2	B3	B4	B5
U1	1	1	-	1	-
U2	1	-	1	1	-
U3	1	1	-	-	1
U4	1	1	-	1	-
U5	-	1	1	-	1

Less Mature

	B1	B2	B3	B4	B5
U1	1	1	1	1	1
U2	1	1	1	1	1
U3	1	1	1	1	1
U4	1	1	1	1	1
U5	1	1	1	1	1

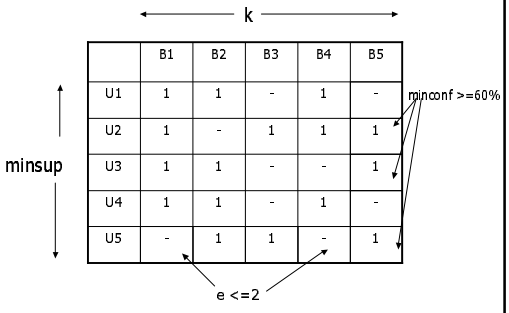
More Mature

- Usually can find longer rules than normal association rules. Need not lower support to do that.
- May find less mature rules.

Finding ETAR.

- To find ETAR, first find ET frequent itemset.
- Definition of ET frequent itemset
  - A group of items I such that
    - contain at least k items
    - there exists a group of users U, |U| > minsup, such that each users in U buy at least (k-e) of the items in I.
    - each item in I is brought by at least minconf% of the users in U.

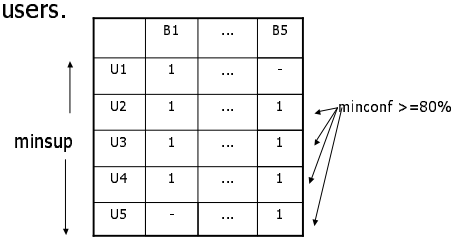
ET Frequent Itemset (Eg.)



	B1	B2	B3	B4	B5
U1	1	1	-	1	-
U2	1	-	1	1	1
U3	1	1	-	-	1
U4	1	1	-	1	-
U5	-	1	1	-	1

Finding ET Frequent Itemset(I)

- Two items can exist in the same ETFI only if they coexist in (2*α - 1)*minsup users.



Finding ET Frequent Itemset(II)

- ETFI also follow anti-monotone property i.e. if the subset of an ETFI does not satisfy minsup then its superset also do not satisfy minsup.
- Proof Sketch: Let I={B1,B2,B3,B4} and e=1. Any user that match any 3 or more items from I will contribute towards supp(I). Consider superset J= {B1,B2,B3,B4, B5} , any user that match 4 or more items from J will contribute towards supp(J).

  But any user that match at least 4 items from J will definitely match at least 3 items from I, thus supp(J) <= supp(I).

Algorithm for finding ETFI

- Generate all candidate with (e+1) items by using first property i.e. all items in the candidate must coexist in at (2*α - 1)*minsup users.
- Run modified Apriori that do partial matching. Keep track for each candidate, the count for each item in the candidate if partial matching is successful.

Generating Rules from ETFI(I)

- Given {B1,B2,B3,B4,B5}, k=5, e=2. Then we generate 5 rules as follow:

  (Any 3 of {B1,B2,B3,B4,B5}) -> B1
  (Any 3 of {B1,B2,B3,B4,B5}) -> B2
  (Any 3 of {B1,B2,B3,B4,B5}) -> B3
  (Any 3 of {B1,B2,B3,B4,B5}) -> B4
  (Any 3 of {B1,B2,B3,B4,B5}) -> B5

- Computing confidence:

	B1	B2	B3	B4	B5
U1	1	1	-	1	-
U2	1	-	1	1	-
U3	1	1	-	-	1
U4	1	1	-	1	-
U5	-	1	1	-	1

Using ETAR

- Two aim when recommending items.
  - (i) Make user buy the recommended items.
  - (ii) Find out more about the user's preferences.
- To achieve objective (i), look for ETAR which are completely matched and recommend higher items with highest confidence.
- To achieve object (ii), look for ETAR which are about to be triggered.

Conclusion

- The problem of collaborative filtering can be tackled in two directions: row clustering or column clustering.
- We point out the problem of performing row clustering and select association rule mining, a form of column clustering to solve the problem.
- To cater to the problem of "immature" data, we proposed introduction error-tolerance into association rule mining.
- Unfortunately, testing are yet to be done.