Data Mining

- Knowledge Discovery in Databases (KDD)
  - Predict or discover rules
    - By analyzing data or detecting patterns
  - Applications
    - Strategy for displaying goods in shops
    - Planning for bargain sales
    - Shipping direct mails
    - Classifying desirable or bad customers
    - and so on

Methods for Data Mining

- Discovering Patterns
  - **Association Rules** (Rakesh Agrawal's group in IBM)
    - **Apriori Algorithm**
    - Example: Basket Analysis (Receipt Analysis), Research Mining, Web Log Mining, …
  - Discovering Sequential Patterns
- Similar Time Sequence
- Clustering
  - Using Decision Trees
  - Using Neural Networks
  - Using Genetic Algorithm
- Statistical Analysis
- …

What is Association Rules?

- Let $I = \{i_1, i_2, \ldots, i_m\}$ be a set of items
- Let $D = \{t_1, t_2, \ldots, t_n\}$ be a set of transactions
  - Each transaction t_i is a set of items such that $t_i \subset I$
- An **association rule** $X \Rightarrow Y$ where $X, Y \in$ I, $X \cap Y = \phi$
  - having two measures of values, **support** and **confidence**
- An **itemset** X has **support** s in the transaction set D
  - s% transactions in D contains X
    - $s \equiv sup(X)$
- A confidence c of $X \Rightarrow Y$ in D means
  - c% of transactions in D that contain X also contain Y
    - c ≡ sup(X, Y)/sup(X) [Conditional Probability of Y, given X]

Mining Association Rules

- A Goal
  - Find all association rules satisfying user specified minimum support and minimum confidence.
- Procedure: **Apriori algorithm**
  1. Derive all large itemsets in which the item satisfy the minimum support.
  2. Using the large itemsets, generate association rules satisfying minimum confidence.

Apriori Algorithm (1)

I. Generate a **large 1-itemset**:
1. Scan the fact database *D*
2. Count occurrence of each item
3. Calculate support for each item
4. Derive a set of items satisfying the minimum support
   - The derived itemset is called a large 1-itemset
     - the 1-itemset contains 1 length items

II. Generate a **candidate 2-itemset**:
1. Select two items from the large 1-itemset
2. Generate all combinations of items in the large 1-itemset.
   - The combination is called a candidate 2-itemset
     - the candidate 2-itemset contains 2 length items

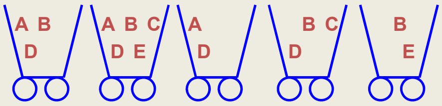
Apriori Algorithm (2)

III. Generate a large 2-itemset:
1. Scan the fact database again
2. Calculate support for each item in the candidate 2-itemset
3. Generate a large 2-itemset satisfying the min support
   - the large 2-itemset contains 2 length items

IV. Continue
1. Generate a candidate *k*-itemset from the large (*k*-1)-itemset
   - the candidate *k*-itemset contains *k* length items
2. Choose the large *k*-itemset
3. Until the large *k*-itemset becomes empty

An Example of Basket Analysis

- Consider the following five baskets (carts)



- Assumptions:
  – minimum support = 0.5
  – minimum confidence = 0.8

Apply the Apriori Algorithm

1. Derive support for each item
   – A: 3/5 = 0.6, B: 0.8, C: 0.4, D: 0.8, E: 0.4
2. Generate the Large 1-itemset: {A, B, D}
3. Generate the candidate 2-itemset: {AB, AD, BD}
4. Derive support for the candidate 2-itemset
   – AB: 2/5=0.4, AD: 0.6, BD: 0.6
5. Generate the large 2-itemset: {AD, BD}
6. Derive confidence
   – AD/A : 3/3 = 1, AD/D: 3/4=0.75, BD/B: 3/4=0.75, BD/D: 3/4=0.75
7. Association rule: A $\Rightarrow$ D,
8. Generate the candidate 3-itemset: {ABD}
9. Derive support for the candidate 3-itemset
   – ABD: 2/5=0.4

Explain the Algorithm in SQL (1)

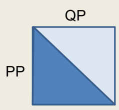
- Generate a large 1-itemset

```
INSERT  INTO LargeItemset1
SELECT  Product_ID
FROM    Fact_Table
GROUP   BY Product_ID
HAVING  COUNT(Transaction_ID) > MSC
```

  - Here, MSC = COUNT(DISTINCT TID) × minimum_support

- Generate a candidate 2-itemset

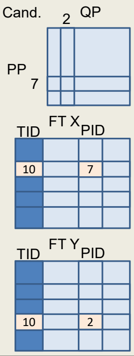
```
INSERT  INTO CandidateItemset2
SELECT  P.Product_ID AS PP, Q.Product_ID AS QP
FROM    LargeItemset1 AS P, LargeItemset1 AS Q
WHERE   PP < QP
```



Explain the Algorithm in SQL (2)

- Generate a large 2-itemset

```
INSERT  INTO LargeItemset2
SELECT  X.Product_ID AS XP, Y.Product_ID AS YP
FROM    Fact_Table AS X, Fact_Table AS Y
        CandidateItemset2 AS C
WHERE   X.Transaction_ID = Y.Transaction_ID
AND     XP = C.PP
AND     YP = C.QP
GROUP   BY XP, YP
HAVING  COUNT(X.Transaction_ID) > MSC
```



2019/6/27 Advance Data Engineering (©H.Yokota)

Sequential Pattern Mining

- Handle not just combinations but sequences
- Sequence database
  - Each sequence has SID
  - (ce) indicate c and e occur at the same time
  - <b(ce)> is a subsequence of <(bf)(ce)b(fg)>

SID	Sequence
10	<(bc)cd(ab)>
20	<(bf)(ce)b(fg)>
30	<(ah)(bf)abf>
40	<(be)(ce)d>
50	<a(bd)bcd(ade)>

- Sequential Pattern Mining:
  - Given min_sup = 2, <b(ce)> is a sequential pattern

2019/6/27 Advance Data Engineering (©H.Yokota) 85

Apriori Property Based Approach

- Apriori property for sequential patterns
  - For a subsequence Sy of a sequence Sx
  - If Sy is not frequent, then Sx is not frequent either
  - For instance
    - <hb> is infrequent, so do <hab> and <(ah)b>
- Generate subsequence DB satisfy min_sup
  - Finding length-1 pattern
  - Generate candidate pattern
  - …

SID	Sequence
10	<(bc)cd(ab)>
20	<(bf)(ce)b(fg)>
30	<(ah)(bf)abf>
40	<(be)(ce)d>
50	<a(bd)bcd(ade)>

2019/6/27 Advance Data Engineering (©H.Yokota) 86

PrefixSpan

- <ab> is prefix of sequence <(ah)(bf)abf> and <a(bd)bcb(ade)>, but not others
- For the prefix <ab>, <(_h)(_f)abf> and <_(_d)bcd(ade)> are prefix-based projection
- Find length-1 sequence patterns at first:
  <a>, <b>, <c>, <d>, <e>, <f>, <g>
  to generate their projection databases
- Then all length-2 sequence patterns
  <aa>, <ab>, <ac>, <ad>, <ae>, <af>, <ag>
  to generate their projection databases

2019/6/27 Advance Data Engineering (©H.Yokota) 87

Efficiency of PrefixSpan

- No candidate sequence needs to be generated
- Projection databases keep shrinking

2019/6/27 Advance Data Engineering (©H.Yokota) 88

Closed Sequential Pattern

- Inclusion of sequence
  - <(bd)cd(ab)> inc <bcda>
  - <(bd)cd(ab)> inc <bd>
  - <bcda> inc <bd>
- Closed Pattern:
  - There is no pattern P', where P inc P' and sup(P) = sup(P')
  - For example, sup(<ab(cd)>) = 3, sup(<abc>) = 3 and <ab(cd)> inc <abc>, so <abc> in not closed
- Can reduce redundancy

SID	Sequence
10	<(bc)cd(ab)>
20	<(bf)(ce)b(fg)>
30	<(ah)(bf)abf>
40	<(be)(ce)d>
50	<a(bd)bcd(ade)>

2019/6/27 Advance Data Engineering (©H.Yokota) 89

Other Pattern Mining Algorithms

- A large number of algorithms have been proposed
  - Pattern-growth methods: FreeSpan
  - Vertical format based mining: SPADE
  - Constraint-based mining: SPIRIT
  - Mining closed sequential patterns: CloSpan, BIDE
  - …
- This course does not focus on details of them
  - Focus on bases of Data Engineering