Penambangan Data [Data Mining]

Kode : SIT5255
Bobot : 2 SKS
Dosen Pengasuh : Dr. Heny Pratiwi, S.Kom., M.Pd., M.TI

Mining Frequent patterns, Associations & Correlations

m set:- set of items.

ample- {computer, printer, MS office software} is 3- item set.

{ milk, bread} is 2-item set.

similarly set of K items is called k-item set.

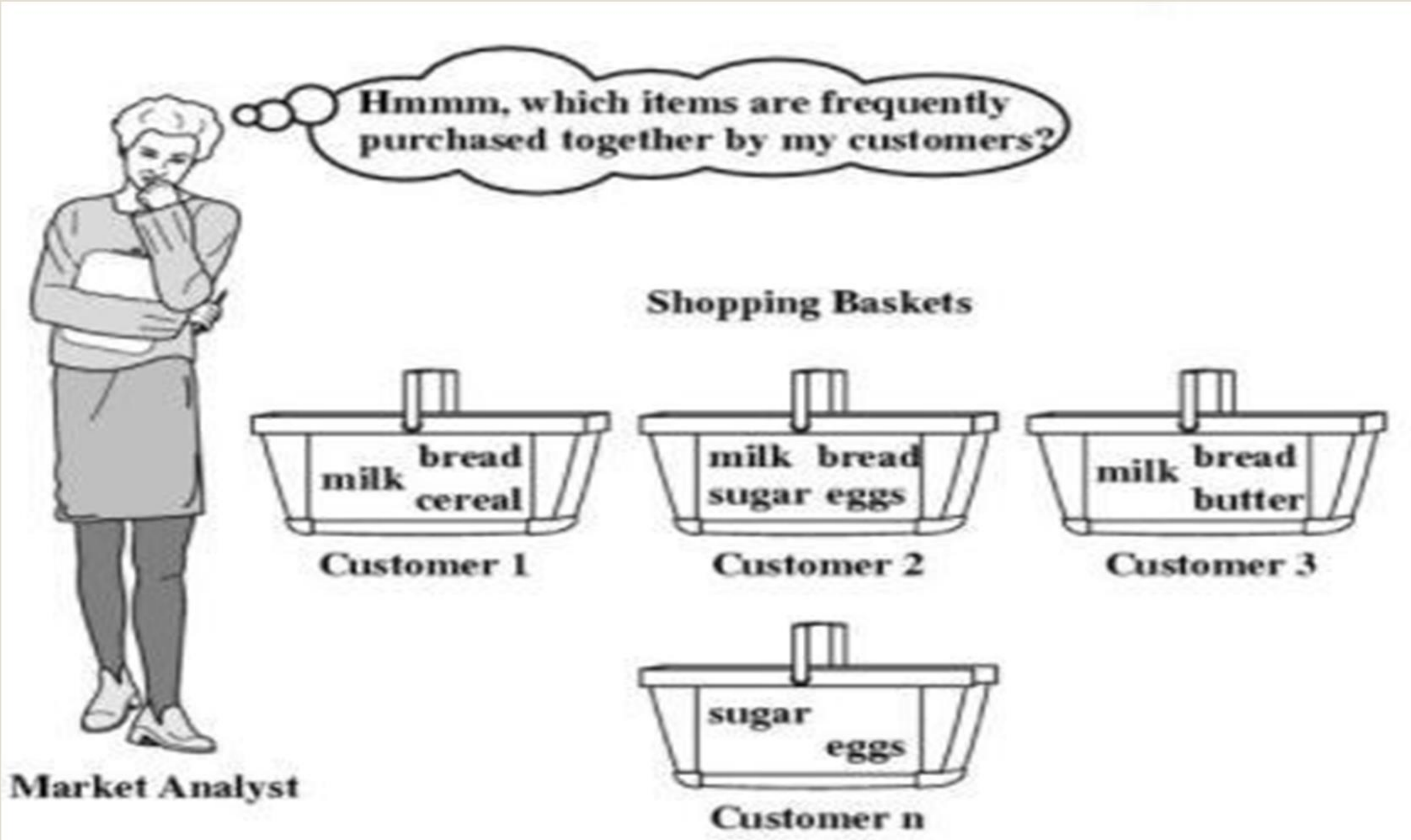
equent patterns are patterns that appear frequently in a data set. Patterns may be msets, subsequences or substructures.

Example: A set of items, such as Milk & Butter that appear together in saction data set. (Also called **Frequent Item set**).

equent item set mining leads to the discovery of associations and correlation nong items in large transactional (or) relational data sets.

his helps in many business decision- making processes like Catalog design, an stomer shopping behavior analysis, etc.

arket Basket Analysis: This is the example of frequent item set mining. This rocess analyzes customer buying habits by finding associations between fferent items that customer places in their shopping baskets.



etailers can use the result by placing the items that are frequently purchase ogether in proximity to further encourage the combined sale of such items.

n our example(in the figure), Milk and bread is frequent, so it can be kept in ximity.

her example is, if customers who purchase computers also tend to bu nter at the same time, then placing the hardware display close to the print y increase the sale of both the items.

<u>ociation rules:</u>

t I= { I_1, I_2, $I_{3,}$.........,I_m } be an item set.

D= { T_1, T_2, $T_{3,}$.........,T_n} be a set of n transactions where each transaction on- empty item set such that T ⊑ I.

[or]

for each i, $T_i \neq \Phi$ and $T_i \sqsubseteq I$

A and B are set of items.

[ex- A= { I_1, I_3, $I_{7,}I_8$ } and B= { I_4, I_5, I_6}]

Association rule is an implication of the form

$$A \Longrightarrow B$$

where $A \subset I$, $B \subset I$, $A \neq \Phi$ and $B \neq \Phi$ & $A \cap B \neq \Phi$

rule $A \Longrightarrow B$ holds in the transaction set D with **Support** s and **Confiden**

pport: This is the percentage of transaction in D that contain A⋃B. Here B means every item in A and every item in B. Support is also written ⋃B). It is also called **Relative support**.

[Note: (A⋃B) ≠ A or B]

herefore,

$$\text{Support } (A \Longrightarrow B) = P(A \cup B).$$

nfidence: This is the percentage of transactions in D containing A that also ntain B. It is also written as P(B/A).

$$\text{Confidence}(A \Longrightarrow B) = P(B/A).$$

$$= \frac{\text{support}(A \cup B)}{\text{support}(A)}$$

$$= \frac{\text{support count } (A \cup B)}{\text{support count } (A)}$$

pport count or Frequency: Number of transactions that contain the item . It is also called **Absolute support**.

Any association rules that satisfy both a minimum suppor threshold(min_sup) and minimum confidence threshold (min_conf) ar called **strong** association.

We have seen in the previous slide that the confidence can easily be derive from the support counts. i.e. If support counts of A, B and A⋃B are found hen we can derive corresponding association rules A ⟹ B and B ⟹ A an heck whether they are strong or not.

Hence mining association rules can be viewed as a two step process:

Finding all frequent item sets and

Generate strong association rules from the frequent item sets.

lote: frequent item set are those item sets that satisfies the min_sup]

losed Frequent item set: An itemset X is **closed** in a data set D if there exist o proper super-itemset Y such that Y has the same support count as X in D.

An itemset X is a **closed frequent itemset** in data set D if X is both sed and frequent.

ximal Frequent itemset: An itemset X is a maximal frequent itemset in a a set D if X is frequent and there exist no super-itemset Y such that $X \subset Y$ an frequent in D.

mple: Let $T_1 = (a_1,a_2,a_3,a_4,a_5)$ and $T_2 = (a_1,a_2,a_3)$

nd minimum support count threshold min_sup=1

erefore, **Set of closed frequent itemset** C= { $\{a_1,a_2,a_3\}$= 2; $\{a_1,a_2,a_3,a_4,a_5\}$=1}.

d **Set of maximal frequent itemset** M= {$\{a_1,a_2,a_3,a_4,a_5\}$=1}.

iori algorithm: (For finding frequent itemsets)

an iterative approach where k-itemsets are used to explore (k+1) itemsets.

os:

e set of frequent 1-itemset is found by scanning the data base and selecting se whose support count satisfy the minimum support. And denote this set as L_1.

L_1 is used to find set of frequent 2-itemset say L_2 .

rther L_2 is used to find L_3 and so on until no more frequent k-itemset can be nd.

te: the finding of each L_k requires one full scan of the database.]

ding L_k (k>=2) :

Join step:

{ Assumption: 1. itemsets are sorted in lexicographic order.

2. l_i [j] means j^{th} item in l_i .

}

he join ($L_{k-1} \bowtie L_{k-1}$) (say it C_k) is performed where members of L_{k-1} are joinab
heir first (k-2) items are in common.

Members l_1 and l_2 of L_{k-1} are joined **if** ($l_1[1] = l_2[1]$ ^ $l_1[2]= l_2[2]$ ^ $l_1[3]= l_2[3]$
………… ^ $l_1[k-2]= l_2[k-2]$ ^ $l_1[k-1] < l_2[k-1]$)

[condition $l_1[k-1] < l_2[k-1]$ ensures no duplicity]

erefore, resulting itemset formed by joining l_1 and l_2 is {$l_1[1]$, $l_1[2]$, $l_1[3]$, $l_1[k$

$l_2[k$

mple:

Let L_2 = [{I_1, I_2}, {I_1, I_3}, {I_1, I_5}]

then,

$L_2 \bowtie L_2$ (i.e. C_3) = [{I_1, I_2, I_3}, {I_1, I_2, I_5},{I_1, I_3, I_5}]

prune step:

he support count of each itemset in C_k is calculated and determine L_k by tting all those itemsets which satisfy the min_sup in C_k .

ote: To determine the support count of each candidate in C_k a complet tabase scan is needed. Therefore to reduce the size of C_k the **Aprio operty** is used.

riori property: if an itemset I does not satisfy the minimum suppo reshold then (I ∪ A) also will not satisfy the min_sup.]

erefore if any (k-1) subset of a candidate k-itemset is not in L_{k-1} , then th ndidate can't be frequent (i.e. does not satisfy min_sup) hence can b moved from C_k .

Example:

nsider the following dataset and for this we have to find frequent itemsets d also have to generate association rules for them

t min_sup = 2

TID	List of items_IDs
T1	I1,I2,I5
T2	I2,I4
T3	I2,I3
T4	I1,I2,I4
T5	I1,I3
T6	I2,I3
T7	I1,I3
T8	I1,I2,I3,I5
T9	I1,I2,I3

Transactional Dataset D

ep 1: create a table C1 that contain support count of each item present in e dataset D.

1

Itemset	Support count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

Now, Compare candidate support count with minimum support count. This gives itemset L1.

L1

Itemset	Support count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

p 2: Generate C2 candidates from L1 (join step), and scan D for count of eac
didate.

2

Itemset	Support count
{I1,I2}	4
{I1,I3}	4
{I1,I4}	1
{I1,I5}	2
{I2,I3}	4
{I2,I4}	2
{I2,I5}	2
{I3,I4}	0
{I3,I5}	1
{I4,I5}	0

Compare candidate support count with minimum support count. This gives itemset L2.

L2

Itemset	Support count
{I1,I2}	4
{I1,I3}	4
{I1,I5}	2
{I2,I3}	4
{I2,I4}	2
{I2,I5}	2

p 3: Generate candidate set C3 using L2 (join step). And scan D for count of ch candidate.

$\bowtie L_2$

{I1,I2,I3}
{I1,I2,I5}
{I1,I3,I5}
{I2,I3,I4}
{I2,I4,I5}
{I2,I3,I5}

But, using Apriori property we can remove {I1, I3, I5},{I2, I3, I4},{I2, I4, I5} and {I2, I3, I5} because every subsets of these sets are not frequent.

Example- for itemset {I1,I3,I5} subset {I3,I5} is not frequent. And for {I2, I3, I4} subset {I3, I4} is not frequent.

erefore,

C3

Itemset	Support count
{I1,I2,I3}	2
{I1,I2,I5}	2

Compare candidate support count with minimum support count. This gives itemset L3.

L3

Itemset	Support count
{I1,I2,I3}	2
{I1,I2,I5}	2

ep 4: Generate candidate set C4 using L3 (join step). And scan D for count ach candidate.

⋈ L_3 | {I1,I2,I3,I5}

refore C4 = Φ

cause the subset {I1, I3, I5} of itemset {I1, I2, I3, I5} is not frequent so there o itemset in C4.

nce algorithm terminated .

we have discovered all the frequent item-sets.

ext lecture we will see the generation of strong association rules and udocode for Apriori algorithm.

Sekian & Terima Kasih