Data Mining Unit 2 - Unit 2

Data Mining And Analytics (SRM Institute of Science and Technology)

Data Mining

Unit-2

* Mining frequent Patterns :-

* Frequent patterns are patterns that appear frequently in a dataset. Set of all such item is called frequent itemset.

* It was first proposed by Agarwal, Imielinski and Swami.

*Sequential frequent pattern :- Buying a digital camera and then memory card if it occurs frequently is called frequent Sequential pattern.

* Market Basket Analysis :-

Frequent & itemset mining leads to discovery of association and corelation among itemset in large transactional dataset.
The process of analyzing customer habbits by finding association between items and customer is placed in the ~~sto~~ shopping basket.

$C_1 \longrightarrow$ Milk, Bread, Cereals
$C_2 \longrightarrow$ Milk, Bread, Sugar, Eggs
$C_3 \longrightarrow$ Milk, Bread, butter
$C_4 \longrightarrow$ Sugar, Egg.

Itemset → {Milk, Bread, Cereal, Sugar, Egg, Butter}

Frequent Itemset → {Milk, Bread, Sugar, Egg}

* Association between any two item is represented in the form of association rules. For example, the information that customer who ~~buck~~ buy computer also buy antivirus at same time is represented in association rule as

Computer → antivirus [support = 2%, Confidence = 60%]

In General: A → B [Support, Confidence]

Support = A ∪ B [A & B together]

Confidence = A/B [Buyed A also buyed B]

* The frequent itemset of k item is denoted by L_k.

* Threshold :- Minimum Support & Confidence above which association can be said interesting.

* Formula

Let $I = [I_1, I_2 -- I_n]$ be a set of items. Let D be a set of database transaction where each transaction T is a set of items such that $T \subseteq I$. Each transaction is associated with an identifier TID. Let A be set of items where A

transaction T is said to contain A iff $A \subseteq T$.

An association rule is implication of form
$A \rightarrow B$ where $A \subseteq I$, $B \subset I$, $A \cap B = \phi$

$$Support(A \rightarrow B) = P(A \cup B)$$

$$Confidence(A \rightarrow B) = P(A|B).$$

Rules that satisfy both minimum support & confidence are called strong.

* Steps :-
Association rule mining can be viewed as 2 step process :-
(i) Find all frequent subset with min_support count.
(ii) Generate Association rule from frequent itemset.

* Problem
Drawback :- It often generate huge number of itemset satisfying minimum support threshold.
For example, a frequent set of length 100 contains
$\binom{100}{1}$ frequent itemset of 1 item
$\binom{100}{2}$ frequent itemset of 2 item.
!
and so on

$$\Rightarrow \binom{100}{1} + \binom{100}{2} + \binom{100}{3} + \cdots \binom{100}{100} = 2^{100} - 1$$
$$\approx 1.27 \times 10^{30}$$

* Solution.

This is too huge number of itemset for any computer to compute or store. To overcome this difficulty we introduce the concept of ~~frequent~~ closed frequent itemset and maximal frequent itemset.

* An itemset X is closed in a data set S if there exist no proper super-itemset Y such that Y contain same support count as X.

* An itemset X is closed frequent itemset in a set S if X is both closed & frequent.

* An itemset X is a maximal frequent itemset in a set S if X is frequent and there exist no super-itemset Y such that X ⊂ Y and Y is frequent in S.

* APRIORI Algorithm :- Finding frequent Itemset Using Candidate Generation

TID	List of Items
T_{100}	I_1, I_2, I_5
T_{200}	I_2, I_4
T_{300}	I_2, I_3
T_{400}	I_1, I_2, I_4
T_{500}	I_1, I_3
T_{600}	I_2, I_5
T_{700}	I_1, I_5
T_{800}	I_1, I_2, I_3, I_5
T_{900}	I_1, I_2, I_3

Minimum_Support = 2.

C_1

Itemset	Support_Count
$\{I_1\}$	6
$\{I_2\}$	7
$\{I_3\}$	6
$\{I_4\}$	2
$\{I_5\}$	2

$\Rightarrow$ L_1

Itemset	Support_Count
$\{I_1\}$	6
$\{I_2\}$	7
$\{I_3\}$	6
$\{I_4\}$	2
$\{I_5\}$	2

Now, we will generate C_2 by joining $L_k \bowtie L_k$
This Join is possible when we have k-1 item is common.

C_2 :-

Itemset	Support_Count
$\{I_1, I_2\}$	4
$\{I_2, I_3\}$	4
$\{I_1, I_4\}$	1
$\{I_1, I_5\}$	2
$\{I_2, I_3\}$	4
$\{I_2, I_4\}$	2
$\{I_2, I_5\}$	2
$\{I_3, I_4\}$	0
$\{I_3, I_5\}$	1
$\{I_4, I_5\}$	0

Here pruning is performed (Removal of Id whose Sup_Count < min_Sup)

Itemset	Sup_Count
$\{I_1, I_2\}$	4
$\{I_1, I_3\}$	4
$\{I_1, I_5\}$	2
$\{I_2, I_3\}$	4
$\{I_1, I_4\}$	2
$\{I_2, I_5\}$	2

Now, we generate C_3 with $L_2 \bowtie L_2$.

C_3 :-

Itemset	Support_Count
$\{I_1, I_2, I_3\}$	2
$\{I_1, I_2, I_5\}$	2
$\{I_1, I_2, I_4\}$	1
$\{I_1, I_3, I_5\}$	1
$\{I_2, I_3, I_4\}$	0
$\{I_2, I_3, I_5\}$	1
$\{I_2, I_4, I_5\}$	0

on pruning ↓

L_3 :-

Itemset	Support_Count
$\{I_1, I_2, I_3\}$	2
$\{I_1, I_2, I_5\}$	2

On again performing $L_3 \bowtie L_3$ to get C_4.

C_4 :-

Itemset	Support_Count
$\{I_1, I_2, I_3, I_5\}$	1

L_4 :- ___ ___

Q Why it is called Apriori?

⇒ Because we are utilizing prior information at each step.

Apriori Property :- All non-empty subset of a frequent itemset must also be frequent.

This method is pruning & property is Apriori property. This property belong to special category of property called antimonotone in sense that if if a set cannot pass a test, all its subset can also not pass it.

Problem :- We basically have to scan entire database again and again. (frequent scanning).

Generating Association Rules :-

$$\text{Confidence} = P(B/A) = \frac{\text{Support}(A \cup B)}{\text{Support}(A)}$$

① Suppose the data contain frequent itemset $I = \{I_1, I_2, I_5\}$. What are the association rules that can be generated from I? Threshold = 75%

$$\Rightarrow \times \{I_1, I_2\} \rightarrow \{I_3\} = \frac{\text{Support}(A \cup B)}{\text{Support}(A)} = \frac{2}{4} = 50\%$$

$$\{I_1, I_5\} \rightarrow \{I_2\} = \frac{2}{2} = 100\%$$

$$\{I_2, I_5\} \rightarrow \{I_1\} = \frac{2}{2} = 100\%$$

$$\times \{I_1\} \rightarrow \{I_2, I_5\} = 2/6 = 33.33\%$$

$$\times \{I_2\} \rightarrow \{I_1, I_5\} = 2/7 \approx 29\%$$

$$\{I_5\} \rightarrow \{I_1, I_2\} = 2/2 = 100\%.$$

$\Rightarrow$ 2nd, 3rd & 6th rule are output.

Improving the efficiency of Apriori Algo:-

1. Hashing technique :- A hash based technique can be used to reduce the size of candidate k-itemsets.

For example :-

$I_1 \rightarrow 1$

$I_2 \rightarrow 2$

$I_3 \rightarrow 3$

$I_4 \rightarrow 4$

$I_5 \rightarrow 5$

For hash function, $h(x,y) = ((\text{order of } x) + (\text{order of } y)) \bmod N$

Index	0	1	2	3	4	5	6
Count	1	2	3	2	3	3	2
Content	$\{I_4, I_5\}$	$\{I_1, I_2\}$ $\{I_2, I_3\}$	$\{I_1, I_2\}$ $\{I_3, I_5\}$ $\{I_4, I_5\}$	$\{I_5, I_4\}$ $\{I_1, I_3\}$	$\{I_3, I_5\}$ $\{I_2, I_3\}$ $\{I_4, I_5\}$	(I_1, I_2) (I_2, I_3) $I_3 I_4$	$\{I_1, I_3\}$

$h(I_1, I_2) = (10 \times (1) + (2)) \bmod 7$

$= 12 \bmod 7$

$= 5$

If min support is 3. Then. items at index 0, 1, 3, 6 can be removed and only items at index 2, 4, 5 will be considered.

2. Transaction Reduction :- It meant reducing the no. of ~~toast~~ transaction scanned in future iteration.

$I_1 \rightarrow 6$ $I_2 \rightarrow 7$ $I_3 \rightarrow 6$ $I_4 \rightarrow 2$, $I_5 \rightarrow 1$

for min sup of 3. we can remove I_4 & I_5 by transaction reduction.

3 Partioning :- It involves partioning the data to find candidate itemsets.

$I_1 = 1$
$I_2 = 3$
$I_3 = 3$
$I_4 = 3$
$I_5 = 2$
$\{I_1, I_5\} = 2$
$\{I_2 I_4\} = 2$
$\{I_4 I_5\} = 1$
$\{I_2 I_3\} = 2$
$\{I_3 I_4\} = 1$
$\{I_2 I_3 I_4\} = 1$

Global

$I_1=1, I_2=1, I_4=1,$
$I_5=1, \{I_1, I_5\}=1$
$\{I_2 I_4\} = 1$

$I_4=1, I_5=1, I_2=1$
$I_3=1, \{I_4 I_5\}=1$
$\{I_2 I_3\} = 1$

$I_5=1, I_2=1, I_3=1, I_4=1$
$\{I_2 I_3 I_4\} = 1.$
$\{I_2 I_3\}=1, \{I_3 I_4\}=1$
$\{I_2 I_4\} = 1$

Local

	I_1	I_2	I_3	I_4	I_5
T_{100}	1	0	0	0	1
T_{200}	0	1	0	1	0
T_{300}	0	0	0	1	1
T_{400}	0	1	1	0	0
T_{500}	0	0	0	0	1
T_{600}	0	1	1	1	0

It consist of 2 phase.

① Partition of data & find local frequent pattern

② Consolidate global frequent pattern.

4 Sampling :- It involves mining on a subset of a given data. The basic idea is to pick a sample from the data and search for frequent itemset in sample instead of scanning entire data.

Size of sample should be that it ~~should~~ can be stored in memory. Because if this does not happen then we will have to store in secondary memory in which time wastage will be more.

FP Growth Algorithm :-

T_1 : { E, K, M, N, O, Y }

T_2 : { D, E, K, N, O, Y }

T_3 : { A, E, K, M }

T_4 : { C, K, M, U, Y }

T_5 : { C, E, I, K, O, O }

Min Support = 3

Step 1 → Generate the frequency Table.

Item	Frequency.
A	1
C	2
D	1
E	4
I	1
K	5
M	3
N	2
O	3
~~R~~	~~1~~
U	1
Y	3

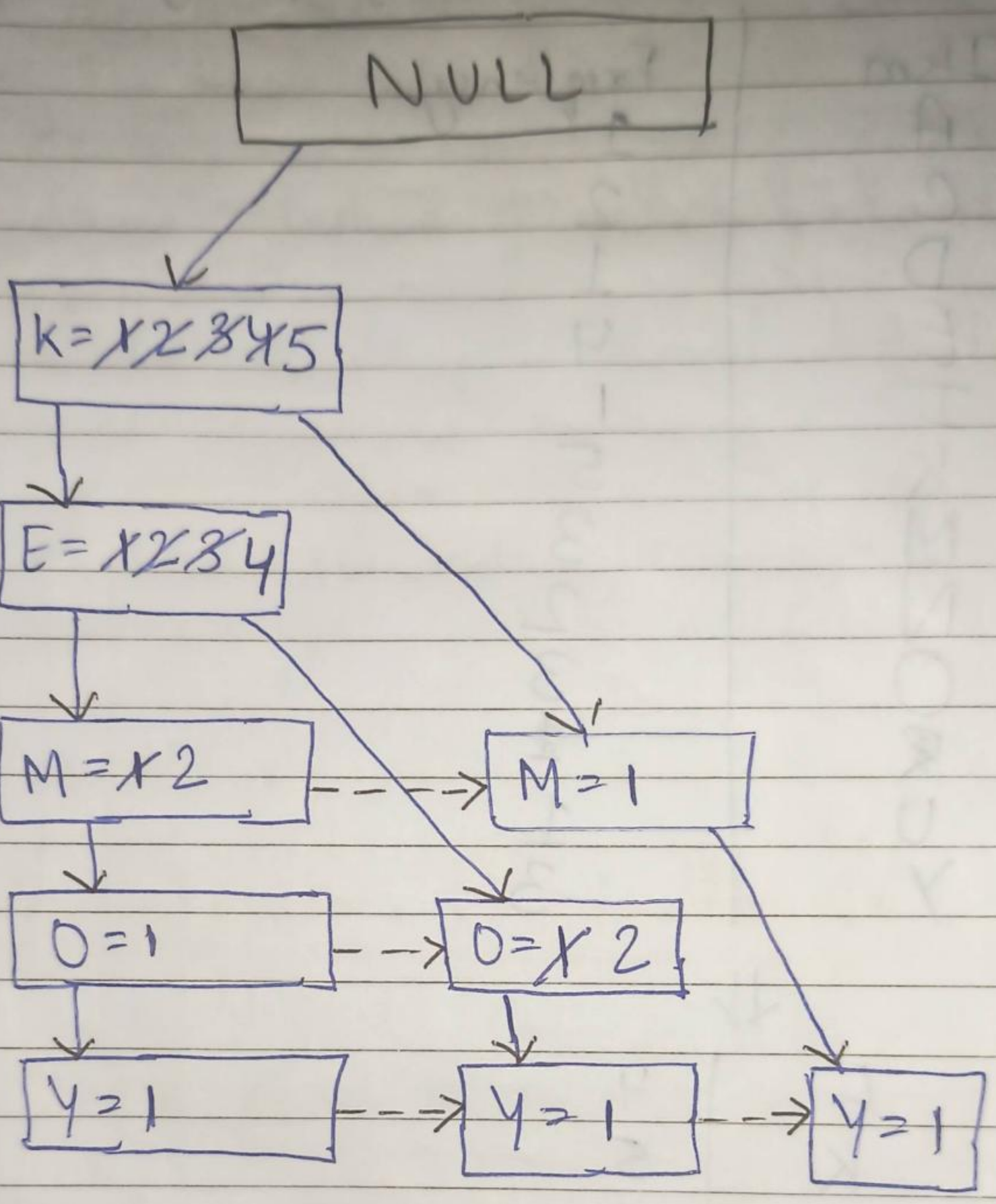
↓

E	4
K	5
M	3
O	3
Y	3

Ordered {K=5, E=4, M=3, O=3, Y=3}

TID	Item Bought	(Ordered) frequent Item
T_1	{E, K, M, N, O, Y}	{K, E, M, O, Y}
T_2	{D, E, K, N, O, Y}	{K, E, O, Y}
T_3	{A, E, K, M}	{K, E, M}
T_4	{C, K, M, U, Y}	{K, M, Y}
T_5	{C, E, I, K, O, D}	{K, E, O}

Step 2 : Draw frequent Pattern Tree



Condition Pattern Base :-

$Y = \{\{K, E, M, O : 1\}, \{K, E, O : 1\}, \{K, M : 1\}\}$

$O = \{\{K, E, M : 1\}, \{K, E : 2\}\}$

$M = \{\{K, E : 2\}, \{K : 1\}\}$

$E = \{\{K : 4\}\}$

$K = \{\}$

Conditional Frequent Pattern Tree

Y: {K:3}

O: {K, E:3}

M: {K:3}

E: {K:4}

Frequent Pattern Generated :-

Y: {K, Y:3}

O: {K, O:3}, {K, E, O:3}, {K, E, O:2}

M: {K, M:3}

E: {K, E:4}

Benifits :- (i) No need to have Candidate Selection after each iteration.

(ii) Complexity Reduce :- No need to scan whole database which was taking $O(n^2)$ time. we just need to scan tree which uses Binary Search $O(\log n)$ time.

* Mining Frequent itemset using Vertical Data format :-

Consider TID and Data set table of Apriori example. That table was based on ~~Frac~~ Transaction. Vertical Data involves Table according to items.

Items	TID Set
I_1	$T_{100}, T_{400}, T_{500}, T_{700}, T_{800}, T_{900}$
I_2	$T_{100}, T_{200}, T_{300}, T_{400}, T_{600}, T_{800}, T_{900}$
I_3	$T_{300}, T_{500}, T_{600}, T_{700}, T_{800}, T_{900}$
I_4	T_{200}, T_{400}
I_5	T_{100} T_{800}

	Itemset	TID Set	Count
	$\{I_1, I_2\}$	T_{100} T_{400} T_{800} T_{900}	4
	$\{I_1, I_3\}$	T_{500} T_{700} T_{800} T_{900}	4
X	$\{I_1, I_4\}$	T_{400}	1
	$\{I_1, I_5\}$	T_{100} T_{800}	2
	$\{I_2, I_3\}$	T_{300} T_{600} T_{800} T_{900}	4
	$\{I_2, I_4\}$	T_{200} T_{400}	2
	$\{I_2, I_5\}$	T_{100} T_{800}	2
X	$\{I_3, I_4\}$		0
X	$\{I_3, I_5\}$	T_{800}	1
X	$\{I_4, I_5\}$		0

Itemset	TID	Count
$\{I_1, I_2 I_3\}$	T_{800}, T_{900}	2
$\{I_1, I_2 I_5\}$	T_{800}, T_{900}	2
~~$\{I_1, I_2, I_4\}$~~		
~~$\{I_1, I_3 I_5\}$~~		
~~$\{I_2, I_3 I_4\}$~~		
~~$\{I_1 I_3 I_4\}$~~		
~~$\{I_2 I_4 I_5\}$~~		
~~$\{I$~~		
$I_1, I_2 I_3 I_5$	——	0

Support, Confidence :- Strong vs Weak Rule

* Strong → If support & Confidence both greater than min_sup & min_Confidence respectively.

* Weak → Otherwise

Q 10,000 transaction.
6,000 Computer game purchase
7500 video game purchase
4000 Both game & video

min_sup = 30%
min_confidence = 60%.

Sol → buy(X, game) → buy(X, video).

Support = P(A ∪ B)

= 4000 / 10000 × 100%

= 40%.

Confidence = P(A/B) = Support(AUB) / Support(A)

= 4000 / 6000 × 100%

= 66.66%

Both greater than it respective threshold

⇒ Strong rules.

However this rule is misleading because probability of purchasing video is 75% which is even larger than confidence (66%). Infact, they ~~pur~~ are negatively associated because purchage of one item actually decreases the likehood of purchasing other.

So, to solve this problem of misleading there must be some other factor in rules alongside support & confidence.

And this is where the concept of correlation came up which is called correlation rules of form

Now, Corelation can be find in multiple ways.

→ A → B [Support, Confidence, Correlation]

1) Lift :- Lift is simple correlation measure that is given as follows :-

$$\text{Lift} = \frac{P(A \cup B)}{P(A)\,P(B)}$$

If value of lift < 1 → negatively correlated

If value of lift = 1 → Event are independent

If value of lift > 1 → positively correlated.

Occurance of one ⇒ occurance of other.

For above example,

$$\text{Lift(game, video)} = \frac{\frac{4000}{10000}}{\frac{6000}{10000} \times \frac{7500}{10000}} = 0.89$$

Lift < 1.

So, there is negative correlation between them.

2) Correlation analysis using χ^2 test

	video	$\overline{\text{video}}$	
game	4000	2000	6000
$\overline{\text{game}}$	3500	500	4000
	7500	2500	10000

$$\text{Expected } (4000) = \frac{6000 \times 7500}{10000} = 4500$$

$$E(2000) = \frac{2500 \times 6000}{10000} = 1500$$

$$E(3500) = \frac{7500 \times 4000}{10000} = 3000$$

$$E(500) = \frac{2500 \times 4000}{10000} = 1000$$

~~game~~	video	$\overline{\text{video}}$
~~video~~ game	4000 (4500)	2000 (1500)
$\overline{\text{game}}$	3500 (3000)	500 (1000)

$$\chi^2 = \sum \frac{(\text{Observed} - \text{Expected})^2}{\text{Expected}}$$

$$= \frac{500^2}{4500} + \frac{500^2}{3000} + \frac{500^2}{1500} + \frac{500^2}{1000}$$

$$= 55.55 + 166.67 + 250 + 83.33$$

$$= 555.55$$

$$= 555.56$$

Because value of χ^2 is greater than 1 and observed value of slot (game, video) = 4000 which is less than expected value of 4500. So, it is negatively correlated.

3. All_Confidence $= \frac{\text{Support}(A \cup B)}{\max(\text{Support}(A), \text{Support}(B))}$

$$= \min(P(B|A), P(A|B))$$

Kulczynski Measure :-

$$\frac{1}{2}\left(P(A|B) + P(B|A)\right).$$

4. Cosine Measure :- $\frac{\text{Support}(A \cup B)}{\sqrt{\text{Support}(A) \times \text{Support}(B)}}$

Max_Confidence $= \max\left(P(B|A), P(A|B)\right)$

$$= \frac{\text{Support}(A \cup B)}{\min(\text{Support}(A), \text{Support}(B))}$$

For our Example of game & video :-

$$\text{all\_conf}(\text{game}, \text{video}) = \frac{4000}{7500}$$

$$= 0.53$$

$$\text{Cosine} = \frac{4000}{\sqrt{7500 \times 6000}}$$

$$= 0.596$$

$$= 0.60$$

Based on rule that it is greater than 0.5 so, it is +vely correlated in allconf & cosine. Since it was -vely correlated in Lift & χ^2. So, good way is to first find it by all_conf & cosine. Then if they are weakly correleated then only other ~~along~~ analysis can be done.