Unit - II

Mining frequent Pattern, Associations and Correlations: Basic Concepts and Methods

Frequent Pattern : The patterns (e.g itemsets, Subsequence or Substructures) that appear frequently in data set.

For ex: a set of items, such as milk and bread, that appear frq. together in a Transaction dataset is a frequent itemset.

A Subsequence such as buying first a PC, then a digital Camera, then a memory card.
→ Frequent sequential pattern

frequent structured pattern → Subgraphs, Sub trees, Sublattice

Example

Market Basket Analysis

"Which groups or set of items are customers likely to purchase on a given trip to the store?".

* It allows retailers to plan which items to put on sale at reduced prices.

association rule ⇒

Computer ⇒ antivirus-software [Support = 2%, Confidence = 60%].

2% of all the transaction under analysis show that computer and antivirus soft. purchased together.

60% probability that customer who bought computer also bought software.

* association rule will be considered interesting if they both satisfy a minimum support threshold and a minimum confidence threshold. These threshold can be a set by users or domain expert.

Frequent itemset, Closed itemset and Association rule

A be a set of items, A transaction T is said to contain A if $A \subseteq T$. let $x = \{I_1, I_2, \ldots I_n\}$ be an itemset.

An association rule is an implication of the form $A \Rightarrow B$, where $A \subset x$, $B \subset x$, $A \neq \phi$, $B \neq \phi$ and $A \cap B = \phi$.

* $A \Rightarrow B$ holds in transaction set D with support s, where s is the percentage of transaction D that contain $A \cup B$.

* $A \Rightarrow B$ has confidence c in transaction set D, where c, is the percentage of transaction in D containing A that also contain B.
This is taken to be the conditional probability $P(B/A)$.

Thus,

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

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

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

$$\text{or} = \frac{\text{Support\_Count}(A \cup B)}{\text{Support\_Count}(A)}$$

Association Rule mining can be viewed as two-step

1. Find all frequent items set.
2. Generate strong association rule from frequent itemsets.

* An itemset X is closed in a dataset D if there exist no proper super-itemset. Such that Y has the same support count as X in D.
* An itemset X is a closed frequent itemset in set D if X is both closed and frequent in D.
* An itemset X is a maximal frequent itemset (or max itemset) in a dataset D if X is frequent and there exists no super-itemset Y such that $X \subset Y$ and Y is frequent in D.

Frequent itemset Mining Method

* **Apriori Algorithm: finding frequent itemsets by Confined Candidate Generation**

→ This ~~itemset~~ Algorithm uses prior knowledge of frequent itemset properties and employs an iterative approach known as level wise search, where K-itemsets are used to explore (K+1) itemsets.

* The set of frequent 1-itemset is founded by scanning the database to accumulate the count for each item and collecting those items that satisfy minimum support.
  The resulting set is denoted by L_1.
* Next, L_1 is used to find L_2. the set of frequent 2-itemsets, which is used to find L_3 and so on. until no more frequent k-itemset can be found.
* The finding of each L_k requires one full scan of the database.
* To improve the efficiency of the level-wise generation of frequent -itemset, property is thus used is called Apriori property.
  → which reduce the search space.

Apriori Property: All nonempty subset of a frequent itemset must also be frequent.

* If a set cannot pass a test, all of its supersets will fail the same ~~test~~ as well ~~as~~, it is called Anti-monotonicity

* L_{k-1} is used to find L_k for $k \geq 2$.
A two step process is followed, consisting of join* and prune* actions:

(i) Join Step

* To find L_k, a set of candidate k-itemsets is generated by joining L_{k-1} with itself.
* let I_1 and I_2 be itemsets in L_{k-1}
* Apriori Assumes items within transaction are sorted in lexicographic order → such that $I_i[1] < I_i[2] < \ldots < I_i[k-1]$.
* The Join is $L_{k-1} \bowtie L_{k-1}$ is performed, where member of L_{k-1} are joinable if their first $(k-2)$ items are in common.
* I_1 and I_2 of L_{k-1} are joind if $(I_1[1] = I_2[1]) \wedge (I_1[2] = I_2[2]) \wedge \ldots \wedge (I_1[k-2] = I_2[k-2]) \wedge (I_1[k-1] = I_2[k-1])$ → This condition ensures that no duplicates are generated.
* The Resulting join I_1 and I_2 is $\{I_1[1], I_2[2], \ldots I_1[k-2], I_1[k-1], I_2[k-1]\}$

② **Prune Step**

* Set of candidate is denoted by C_k.
* C_k is subset of L_k. that is, its member may or may not be ~~member~~ frequent. but all of the frequent k-itemset are included in C_k.

* C_k however, can be huge, and could ~~have~~ involve heavy computation.
* To reduce the size of C_k, the Apriori property is used.

* Any (k-1) itemset that is not frequent cannot be a subset of frequent k-itemset.
Hence, if any (k-1) subset of candidate k-item set is not in L_{k-1}, then the candidate cannot be frequent either and so can be removed from C_k.

Example // Transaction Datas for an AllElectronics

$\frac{22}{9} = 2.2$

TID	List of item-IDs
T100	I_1, I_2, I_5
T200	I_2, I_4
T300	I_2, I_3
T400	I_1, I_2, I_4
T500	I_1, I_3
T600	I_2, I_3
T700	I_1, I_3
T800	I_1, I_2, I_3, I_5
T900	I_1, I_2, I_3

$C_1 \Rightarrow$ 1-itemset

Scan D for count of each candidate →

itemset	Count
I_1	6
I_2	7
I_3	6
I_4	2
I_5	2

min_sup = 2

→ Compare candidate support count with min support count

i.e ⇒ min_sup = 2

L_1

itemset	Sup count
I_1	6
I_2	7
I_3	6
I_4	2
I_5	2

Generate C_2 candidate from L_1 →

C_2 - 2-itemset

itemset
$\{I_1, I_2\}$
$\{I_1, I_3\}$
$\{I_1, I_4\}$
$\{I_1, I_5\}$
$\{I_2, I_3\}$
$\{I_2, I_4\}$
$\{I_2, I_5\}$
$\{I_3, I_4\}$
$\{I_3, I_5\}$
$\{I_4, I_5\}$

Scan D for count → of each candidate

C_2 - 2 itemset

itemset	Sup. count	
$\{I_1, I_2\}$	4	
$\{I_1, I_3\}$	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	×

min sup = 2 →

Compare candidate support count → with minimum support count

as min sup = 2

L_2

itemset	Sup count
I_1, I_2	4
$I_1\ I_3$	4
$I_1\ I_5$	2
$I_2\ I_3$	4
$I_2\ I_4$	2
$I_2\ I_5$	2

Generate C_3 → candidates from L_2

C_3
3-itemset

itemset
$I_1\ I_2\ I_3$
$I_1\ I_2\ I_5$

$I_1 I_2 I_3$ ✓
$I_2 I_3 I_4$ ×
$I_1 I_2 I_5$ ✓
$I_1 I_3 I_5$ ×
$I_2 I_3 I_4$ ×
$I_2 I_3 I_5$ ×
$I_2 I_4 I_5$ ×

Scan D for count of each candidate →

C_3

itemset	Sup cnt
$I_1 I_2 I_3$	2
$I_1 I_2 I_5$	2

Compare candidate support count → with min. support count

L_3

itemset	Sup count
$I_1 I_2 I_3$	2
$I_1\ I_2\ I_5$	2

Now, C_4 → 4-itemset,
which thus, doesn't satisfy for min sup=2 when scan D. It will be pruned as it is not frequent. ← $I_1 I_2 I_3 I_5$

→ Thus C_4 is ϕ. Algo. terminates Here.

* Generating Association Rules from frequent itemsets

* Strong association rules satisfy both minimum support and minimum confidence.

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

Association rules can be generated as follows:

① for each frequent itemset I, generate all nonempty subsets of I.

② for every nonempty subset s of I, output the rule "$s \Rightarrow (I-s)$" if $\frac{\text{Support\_count}(I)}{\text{Support\_count}(s)} \geq \text{min\_conf}$

where, min_conf is the minimum confidence threshold

Example:

frequent itemset in above example are:

$X = \{I_1, I_2, I_5\}$

Association rule for X ⇒

Non-empty subsets of X are $\{\{I_1, I_2\}, \{I_1, I_5\}, \{I_2, I_5\}, \{I_1\}$
$\{I_2\}\ \{I_5\}$

Resulting Association Rule with its confidence ⇒

$\{I_1, I_2\} \Rightarrow I_5$, Confidence $= \frac{2}{4} = 50\%$.

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

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

$I_1 \Rightarrow \{I_2, I_5\}$, confidence $= \frac{2}{6} \Rightarrow 33\%$

$I_2 \Rightarrow \{I_1, I_5\}$, confidence $= \frac{2}{7} = 29\%$

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

Improving Efficiency of Apriori

1. Hash-based Technique :- hashing itemsets into corresponding buckets
2. Transaction Reduction. → Reducing the no. of transaction scanned in future iterations.
3. Partitioning. → partitioning the data to find candidate itemsets
4. Sampling - mining on a subset of given Data.
5. Dynamic itemset counting :- adding candidate itemsets at different points during a scan.

Apriori candidate generate and test method suffer from two non trivial costs:

① It may still need to generate a huge number of candidate sets.
for ex. if there are 10^4 frequent 1-itemsets the Apriori algorithm will need to generate more than 10^7 candidate 2-itemsets.

② It may need to repeatedly scan the whole database and check a large set of candidate by pattern matching. It is costly to go over each transaction in the database to determine the support of candidate itemsets.

So, a design method is needed that mines the complete set of frequent items without such a costly candidate generation process, i.e frequent pattern growth (FP-growth)

FP-Growth — 'Divide and Conquer Strategy'

* first it compress the database representing frequent items into a FP tree, which retains the itemset association information.

* It then divide the compressed database into a set of conditional database, each associated

with one frequent item or "pattern fragement" and mines each database separately.

* for each "pattern fragement" only its associated data sets need to be examined. Therefore, this approach may substantially reduce the size of the data sets to be searched, along with the "growth" of patterns being examined.

Ex **FP-Growth** (Finding frequent pattern (itemsets) without candidate generation)

TID	List of item IDs	
T100	I_1, I_2, I_5	$\to I_2\ I_1\ I_5$
T200	I_2, I_4	$\to I_2\ I_4$
T300	I_2, I_3	$\to I_2\ I_3$
T400	I_1, I_2, I_4	$\to I_2\ I_1\ I_4$
T500	I_1, I_3	$\to I_1\ I_3$
T600	I_2, I_3	$\to I_2\ I_3$
T700	I_1, I_3	$\to I_1\ I_3$
T800	I_1, I_2, I_3, I_5	$\to I_2\ I_1\ I_3\ I_5$
T900	I_1, I_2, I_3	$\to I_2\ I_1\ I_3$

Scan D for count of each itemset

Itemset	Count
I_1	6
I_2	7
I_3	6
I_4	2
I_5	2

let min_support is 2. The set of frequent items is sorted in the order of descending support count. This resulting set or list is denoted by L.

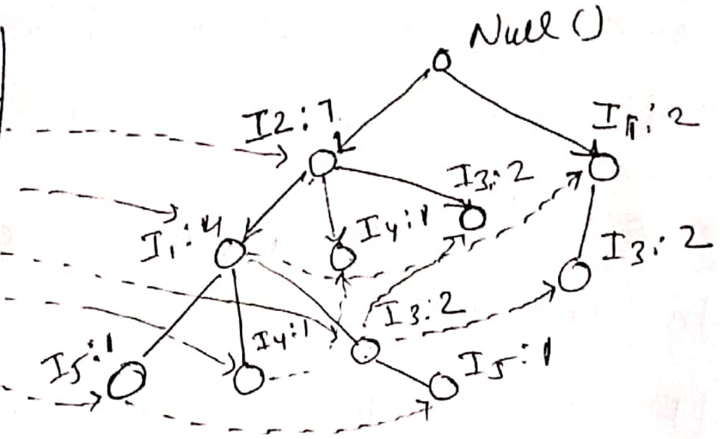
L (1-itemset)

itemset	Count
I_2	7
I_3	6
I_1	6
I_4	2
I_5	2

Now an FP-tree is constructed as follows:

* first create the root of tree, labelled with "null".
* scan database D a second time.

item ID	Support Count	Node link
I_2	7	•
I_1	6	•
I_3	6	•
I_4	2	•
I_5	2	•



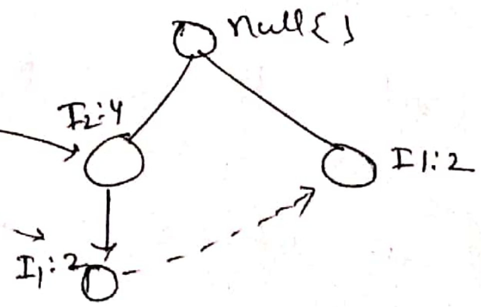
* To faciliate tree traversal, an item header table is built so that each item points to its occurence in the tree via a chain of node links.

* We first consider I_5, which is last item L, rather than the first.

item	Conditional Pattern Base	Conditional FP Tree	Frequent Pattern Generated
I_5	$\{I_2, I_1 :1\}\{I_2, I_1, I_3 :1\}$	$\{I_2:2, I_1:2\}$	$\{I_2 I_5:2\}\{I_1 I_5:2\}\{I_2, I_1, I_5:2\}$
I_4	$\{I_2, I_1 :1\}\{I_2:1\}$	$\{I_2:2\}$	$\{I_2 I_4:2\}$
I_3	$\{I_2, I_1 :2\}\{I_2:2\}\{I_1:2\}$	$\{I_2:4, I_1:2\}\{I_1:2\}$	$\{I_2 I_3:4\}\{I_1 I_3:4\},\{I_2 I_1 I_3:2\}$
I_1	$\{I_2:4\}$	$\{I_2:4\}$	$\{I_2 I_1:4\}$

* As, I_2 is connected to root node only, I_2 is not from any other or we can say I_2 is generated by any other node. So ignore I_2.

Item ID	Count
I_2	4
I_1	4



Ex.2

TID	items	
1	b d c a	→ dabc
2	e d c	→ dc
3	a b	→ ab
4	a c d	→ dac
5	f g d b	→ db

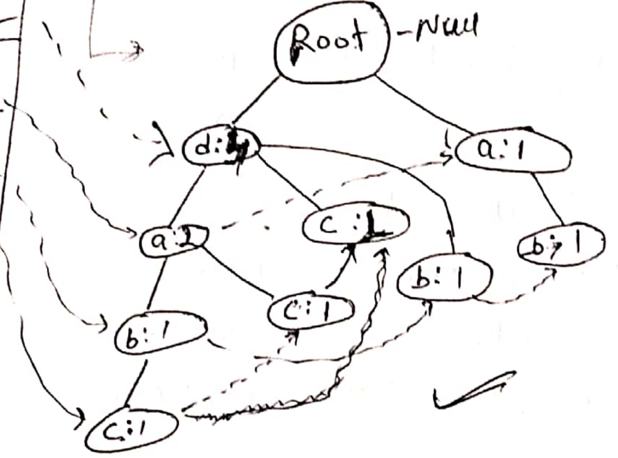
itemset	Count	
a	3	
b	3	
c	3	
d	4	
e	1	×
f	1	×
g	1	×

As, min_support is 2

Arrange according to Decending ordering Count

In Dec. order

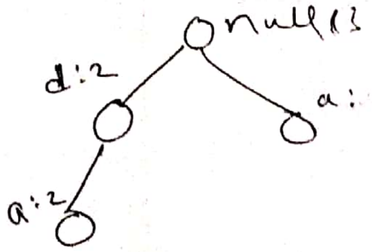
itemset	count	link
d	4	
a	3	
b	3	
c	3	



item	Conditional Pattern Base	Conditional FP Tree	Frequent Pattern Generated
c	{d,a,b:1}{d,a:1}{d:1}	{d, ~~a~~:3, a:2}	{d, c:3, d,a:2}{d,c,a:3}
b	{d,a:1} {d:1} {a:1}	{d:2, a:2}	{d,b:2}{a,b:2}{d,a,b:2}
a	{d:2}	{d:2}	{d, a:2}

* As 'd' is attached to root or null node only, so We can ignore it.

item ID	Count
d	2
a	2



Mining frequent itemsets Using the vertical Data format

* Where item set is an item name. and TID_Set is the set of transaction identifie(r) Containing the item. This is known as vertical format.

Vertical Data format of Transaction Data ⟹

1-itemset

itemset	TID_set
I_1	T100, T400, T500, T700, T800, T900
I_2	T100, T200, T300, T400, T600, T800, T900
I_3	T300, T500, T600, T700, T800, T900
I_4	T200, T400
I_5	T100, T800

2 itemset

itemset	TID_set
$\{I_1 I_2\}$	{T100, T400, T800, T900}
$\{I_1 I_3\}$	{T500, T700, T800, T900}
$\{I_1 I_4\}$	{T400}
$\{I_1 I_5\}$	{T100 T800}
$\{I_2, I_3\}$	{T300, T600, T700, T800, T900}
$\{I_2, I_4\}$	{T200, T400}
$\{I_2, I_5\}$	{T100, T800}
$\{I_3, I_4\}$	{T800}
$\cancel{\{I_4, I_5\}}$	

⟹

itemset	TID_set
$\{I_1 I_2 I_3\}$	T800 T900
$\{I_1, I_2, I_5\}$	T100, T800

Scanned by CamScanner

{Pattern Evaluation Method}

* How can we tell which strong association rules are really interesting ?

Ex. buys(X, "computer games") ⇒ buys(X, "videos")

[Support = 40% , Confidence = 66%]

$\frac{4000}{10000} = 40\%$ $\frac{4000}{6000} = 66\%$

computer games

10000 transaction includes ⇒ 6000 {computer games / videos}

7500 video include ⇒ 4000

A ⇒ B [Support, Confidence, Correlation]

* **lift** is a simple correlation measure that is given as follows:

* The occurrence of itemset A is independent of the occurrence of the itemset B if $P(A \cup B) = P(A)P(B)$ otherwise, itemsets A and B are dependent and correlated as events.

$$\boxed{lift(A, B) = \frac{P(A \cup B)}{P(A)\,P(B)}}$$

if above eq. is < 1, then the occurrence of A is negatively correlated with the occurrence of B.

Correlation analysis using lift

2X2 Contigency Table

	game	$\overline{game}$	Σ_{row}
video	4000 (4500)	3500 (3000)	7500
$\overline{video}$	2000 (1500)	500 (1000)	2500
Σ_{col}	6000	4000	10,000

$$P(game) = \frac{6000}{10000} = 0.60$$

$$P(video) = \frac{7500}{10000} = 0.75$$

$$P(game, video) \Rightarrow lift(game, video) = \frac{P(game \cup video)}{P(game)\ P(video)}$$

$$\Rightarrow \frac{4000}{10000} = 0.40$$

$$\text{lift of rule} \Rightarrow \frac{P\{game, video\}}{P\{game\} \times P\{video\}} = \frac{0.40}{0.60 \times 0.75} = 0.89$$

* Because this value is less than 1, there is a negative correlation between occurrence of {game} and {video}.

Correlation analysis using χ^2

$$\chi^2 = \sum_{i=1}^{n}\sum_{j=1}^{2}\frac{(o_{ij}-e_{ij})^2}{e_{ij}} \Rightarrow \frac{(4000-4500)^2}{4500} + \frac{(2500-3000)^2}{3000}$$

$$+ \frac{(2000-1500)^2}{1500} + \frac{(500-1000)^2}{1000}$$

$$\Rightarrow 55.55 + 83.33 + 166.66 + 250$$

$$\Rightarrow \underline{\underline{555.6}}$$

* Here, χ^2 value is greater than 1.

* the observed value of the slot (game, video = 4000), which is less than value of 4500, (buying game) and (buying video) are negatively correlated.

Comparision of Pattern Evaluation Measures

① All_Confidence ⇒

$$all\_conf(A,B) = \frac{sup(A \cup B)}{max\{sup(A), sup(B)\}} = min\{P(A/B), P(B/A)\}$$

② Max_Confidence ⇒

$$max\_Conf(A,B) = max\{P(A/B), P(B/A)\}$$

(3) Kulczynski Measure

$$\text{kulc}(A,B) = \frac{1}{2}\left(P(A/B) + P(B/A)\right)$$

* also viewed as average of two confidence measure. or average of two conditional prob.

(4) Cosine Measure

$$\text{Cosine}(A,B) = \frac{P(A \cup B)}{\sqrt{P(A) \times P(B)}} = \frac{\text{sup}(A \cup B)}{\sqrt{\text{sup}(A) \times \text{sup}(B)}}$$

$$\Rightarrow \sqrt{P(A/B) \times P(B/A)}$$

* Harmonized lift Measure
* Squareroot is taken on the product of probabilities of A and B.

Example:

	milk	$\overline{milk}$	Σ row
Coffee	mc 10000 (1080)	$\overline{m}c$ 1000 (9919)	c - 11000
$\overline{Coffee}$	$m\overline{c}$ 1000 (9919)	$\overline{m}\overline{c}$ 100,000 (91080)	$\overline{c}$ - 101000
Σ col	m 11000	$\overline{m}$ 101000	Σ = 112000

Six pattern Evaluation Measure using Contigency Table

Data set	mc	$\overline{m}c$	$m\overline{c}$	$\overline{mc}$	χ^2	lift	all Conf	Max Conf	Kulc	Cosine
D_1	10,000	1000	1000	100,000	90557	9.26	0.91	0.91	0.91	0.91
D_2	10,000	1000	1000	100						
D_3	100	1000	1000	100,000						
D_4	1000	1000	1000	100,000						
D_5	1000	100	10000	100,000						
D_6	1000	10	100,000	100,000						

$$\chi^2(D_1) = \frac{(10000-1080)^2}{1080} + \frac{(1000-9919)^2}{9919} + \frac{(1000-9919)^2}{9919} + \frac{(100000-91080)^2}{91080}$$

$$73672.5 + 8019 + 8019 + 873$$

$$\Rightarrow 73672.5 + 8019 + 8019 + 873$$

$$\Rightarrow 90557$$

$$\text{lift}(D_1) \Rightarrow \frac{P(\text{Coffee} \cup \text{milk})}{P(\text{coffee}) \times P(\text{milk})} = \frac{10000/112000}{\frac{11000}{112000} \times \frac{11000}{112000}} = \cancel{0.0892}$$

$$= \frac{0.0982}{0.0982 \times 0.982}$$

$$= \frac{0.0982}{0.00964} = 9.26$$

$$\text{all\_conf}(D_1) \Rightarrow \min\left(P\left(\frac{\text{coffee}}{\text{milk}}\right), P\left(\frac{\text{milk}}{\text{coffee}}\right)\right)$$

$$\Rightarrow P\left(\frac{\text{coffee}}{\text{milk}}\right) \Rightarrow \frac{P(\text{coffee} \cup \text{milk})}{P(\text{milk})} \Rightarrow \frac{\frac{10000}{\cancel{112000}}}{\frac{11000}{\cancel{112000}}}$$

$$\Rightarrow 0.909 \rightarrow 0.91$$

$$\Rightarrow P\left(\frac{\text{milk}}{\text{coffee}}\right) = \frac{P(\text{coffee} \cup \text{milk})}{P(\text{coffee})} = \frac{\frac{10000}{\cancel{112000}}}{\frac{11000}{\cancel{112000}}}$$

$$\Rightarrow 0.91$$

$$\text{all\_conf}(D_1) \Rightarrow \min(0.91, 0.91)$$

$$= \underline{0.91}$$

$$\text{max\_conf}(D_1) = \max(0.91, 0.91)$$

$$= 0.91$$

$$\text{kulc}(D_1) \Rightarrow \frac{1}{2}\left(P(A/B) + P(B/A)\right)$$

$$\Rightarrow \frac{1}{2}\left[P(\text{coffee}/\text{mik}) + P(\text{milk}/\text{coffee})\right]$$

$$\Rightarrow \frac{1}{2}\left[0.91 + 0.91\right]$$

$$= 0.91$$

$$\text{Cosine} \Rightarrow \sqrt{P(A/B) \times P(B/A)}$$

$$= \sqrt{P\left(\frac{\text{coffee}}{\text{milk}}\right) \times P\left(\frac{\text{milk}}{\text{coffee}}\right)}$$

$$= \sqrt{0.91 \times 0.91}$$

$$= \sqrt{0.8281} = \underline{\underline{0.91}}$$