IFN for Recommender Systems

Ideal Flow Network

Kardi Teknomo, PhD

3-5 October 2022

Robert B. Bank Auditorium
Asian Institute of Technology
Pathumthani, Thailand

Recommender Systems

- Recommender Systems (Market Basket Analysis) is an information filtering system that provide suggestions for items that are most pertinent to a particular user.
  - automate various decision-making processes by providing personal and high quality recommendations
    - what product to purchase
    - what music to listen to
    - what online news to read

Kardi Teknomo, PhD

Market Basket Analysis

- Market basket analysis is a tool of knowledge discovery about co-occurrence of nominal or categorical items.
- MBA is a data mining technique to derive association between two data sets.
- We have categorical data of transaction records as input to the analysis and the output of the analysis is association rules as a new knowledge directly from data.

Kardi Teknomo, PhD

Association Rules

- Form: X -> Y
- Left hand side: antecedent
- Right hand side: consequent
- “people who bought items on set X are often also bought items on set Y”.
- X = {bread, cheese }
- Y = {butter, jam, milk, honey }
- People who bought bread and cheese also bought butter, jam, milk and honey

Support

- **Support** = the frequency of transactions to have the all the items on both sets X and Y are bought together.
- = probability of the intersection of set X and set Y.

$$support\ (X \rightarrow Y) = P(X \cap Y) = \frac{n(X \cap Y)}{N}$$

- **Confidence** = computed as conditional probability to obtain set Y given set X.

$$confidence\ (X \rightarrow Y) = P(Y|X) = \frac{n(X \cap Y)}{n(X)}$$

- = the total frequency of the set intersection divided by the total frequency of set X.

Naive - Exhaustive Approach

- Step by step computation of Market Basket Analysis is as follow:

1. Generate all possible association rules
2. Compute the support and confidence of all possible association rules
3. Apply two thresholds criteria: *minimum support* and *minimum confidence* to obtain the association rules.

Kardi Teknomo, PhD

Generating All Possible A.R.

- the total number of possible association rules, R, is exponential to the number of items, d:

$$R = 3^d - 2^{d+1} + 1$$

(Tan, Steinbach & Kumar, 2006)

Total number of items, d	1	2	3	4	5	10	100	500
Total possible association rules, R	0	2	12	50	180	57002	5.15378E+47	3.636E+238

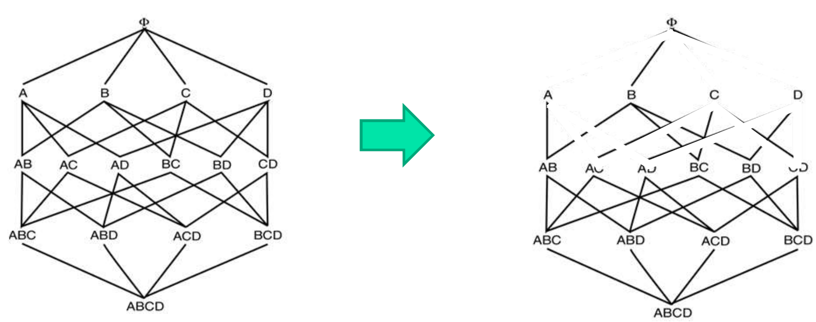
- If d is large, very difficult to generate manually
- Need apriory algorithm to prune the possibilities or use IFN as alternative approach

7

Kardi Teknomo, PhD

State of the Art: Apriori

- Pruning itemset: If there is any itemset which is infrequent, its superset should not be generated
  - If an itemset is not large, none of its supersets are large. ➔ cut based on threshold

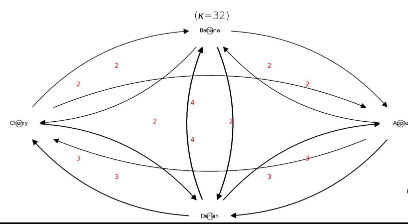


8

Kardi Teknomo, PhD

Proposed: IFN for MBA

- Ideal Flow Network (IFN) is one of the most natural techniques to create a product graph for Recommender Systems.
  - Nodes represent the items
  - Links represent the co-occurrence relationship between items.



9

Kardi Teknomo, PhD

What is IFN

- Ideal Flow Network (IFN) is a steady state relative flow distribution in a strongly connected network where the flows are conserved

IFN = Irreducible Premagic

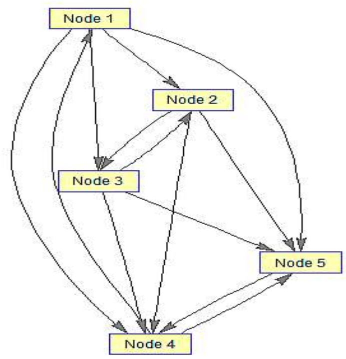
- Represented by network (digraph) or matrix

10

Random Walk on Network

- Given: Strongly connected network
- At each node, select the available link at constant probability distribution
- Record the link flow = sum of trajectories
- The agents keep moving until time T=∞ is achieved.

- Simplified version: edge length = 1, agent speed = 1 (each agent jump from one node to the next node at 1 time step)



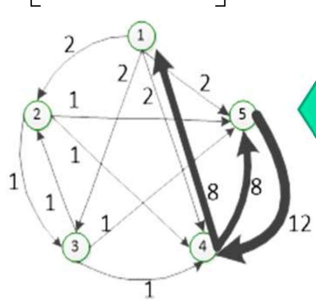
Example

N=200 agents, T=1000 time steps

$$\mathbf{A}=\begin{bmatrix}0&1&1&1&1\\0&0&1&1&1\\0&1&0&1&1\\1&0&0&0&1\\0&0&0&1&0\end{bmatrix}$$

$$flow=\begin{bmatrix}0&985&920&966&945\\0&0&492&450&529\\0&467&0&482&465\\3806&0&0&0&3784\\0&0&0&5709&0\end{bmatrix}$$

$$\begin{array}{cc}\mathbf{F}=\begin{bmatrix}0&2&2&2&2\\0&0&1&1&1\\0&1&0&1&1\\8&0&0&0&8\\0&0&0&12&0\end{bmatrix} & \begin{array}{c}8\\3\\3\\16\\12\end{array}\\ \begin{array}{ccccc}8&3&3&16&12\end{array} & \end{array}$$



Kardi Teknomo, PhD

Interesting Phenomena

- Regardless of the number of agents N or total simulation time T, when $N \cdot T$ is quite large to fill the network, **we have asymptotic values** of relative flow ratio and relative node ratio
  - Flow ratio and node ratio (number of agents' visit on links and nodes) depend on network structure and probabilities to enter each link. They are not depending on simulation time nor number of agents.

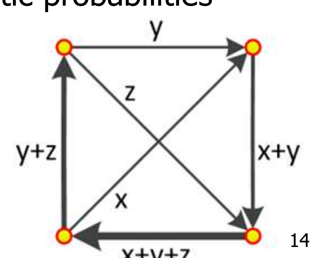
$$\mathbf{F} \cong \lim_{N \cdot T \to \infty} \frac{\mathbf{R}}{\min \mathbf{R}} \text{ if } r_{ij} \neq 0$$

13

Kardi Teknomo, PhD

Characteristics of IFN

1. **Irreducible - Strongly connected network**: from any node can go to any other node and go home
2. **Premagic - Flow preservation**: the link flows are preserved, in each node the total inflow = total outflow
3. **Fully utilized network**: the link flows are all positive
4. **Relative flow**: positive scaling produces equivalence network that preserve the stochastic probabilities



14

IFN Software

- **Python code** is freely available in GitHub:
  - https://github.com/teknomo/IdealFlowNetwork
- **Excel Add Ins** is freely available in Revoledu:
  - https://people.revoledu.com/kardi/research/trajectory/ifn/IFN-Excel-Add-In.html
- **HTML (JavaScript)** version is also freely available in Revoledu ➔ Accessible in Mobile
  - https://people.revoledu.com/kardi/tutorial/IFN/

Kardi Teknomo, PhD

Why use IFN?

- Using **traffic flow network** as modeling metaphors and naming convention make us easier to apply to many other applications
  - Irreducible (strongly connected)
  - Flow
  - Premagic
- As long as the problems at hand can be formulated as flow equilibrium in a strongly connected network graph, we can consider to use IFN as our modeling tools

Kardi Teknomo, PhD

IFN for Data Association

- Node = item
- Link = go together (bi-directional, thus the matrix is symmetric)
  - Scan the data that goes together as additional count in the adjacency matrix. New item means new row and new column in the count matrix.
  - Compute the IFN
  - We can get the association rules simply by setting percentage on the link flow above certain threshold

$$\boldsymbol{\pi} = \begin{bmatrix} \mathbf{S}^T - \boldsymbol{I} \\ \mathbf{j}^T \end{bmatrix} \backslash \begin{bmatrix} 0 \\ \kappa \end{bmatrix} \Rightarrow \mathbf{F} = \boldsymbol{\pi}\mathbf{j}^T \circ \mathbf{S}$$

17

Kardi Teknomo, PhD

IFN-Cycle Theorem

- Suppose we have set of cycles C
- Flow-set element is defined as

$$\tilde{f}_{ij} = \{c \in C | c = (v_1, v_2, \cdots, v_h = i, v_{h+1} = j, \cdots, v_k, v_1)\}$$
$$\exists h, 1 < h < k$$

- Flow matrix element is defined as count of flow-set element

$$f_{ij} = \left|\tilde{f}_{ij}\right|$$

- Then the flow matrix would be premagic and irreducible (= ideal flow matrix)

18

Kardi Teknomo, PhD

The overlay operation

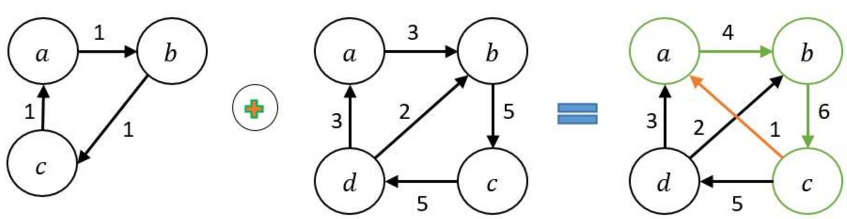
- The overlay operation (symbol $\oplus$ in $\mathbf{F}^s_{k+1} = \mathbf{F}^s_k \oplus \mathbf{F}^a$) of a network $\mathbf{F}^a$ into another network $\mathbf{F}^s$ is done as follow.
  - For each edge $n_1 n_2$ in $\mathbf{F}^a$ with its flow $f_{n_1 n_2}$
    - If $n_1 n_2$ does not exist in $\mathbf{F}^s$, create $\mathbf{F}^s$ with flow $f_{n_1 n_2}$
    - If $n_1 n_2$ already exist in $\mathbf{F}^s$, update the flow of the edge with additional $f_{n_1 n_2}$

19

Kardi Teknomo, PhD

Example of Overlay Operation

- Pivot: in green color
- Expansion: in orange color



20

Algorithm IFN Association

Training of IFN Association to get system IFN based on list of transaction item sets
Given: list of transaction item sets
Output: IFN, $\mathbf{F}^s$
Algorithm: For each transaction-list L: 1. Form a complete graph $\mathbf{G}$ from each item in a transaction L 2. Overlay the complete graph into the IFN to update its flow $\mathbf{F}^s = \mathbf{F}^s \oplus \mathbf{G}$ Return $\mathbf{F}^s$

Kardi Teknomo, PhD

Algorithm IFN Association

Prediction based on IFN Association for a transaction list of set of items
Given transaction-list L
Output: prediction dictionary of item and its flow P
Algorithm: Set Prediction P as empty For each item n_1 in a transaction L: If item n_1 is in IFN $\mathbf{F}^s$: Get the adjacent nodes n_2 and its flow $f_{n_1 n_2}$ in IFN that is not in L Update Prediction based on the adjacent nodes and its flow Return Prediction P

Kardi Teknomo, PhD

Support and Confidence

- Support is computed as the flow of node divided by the total flow in IFN.

$$support(n_i) = \frac{f_{n_1}}{\kappa}$$

- Confidence is computed as the total flow in the direct links divided by total node flow of the node origins.

$$confidence(n_i, n_j) = \frac{\sum_i \sum_j f_{n_i n_j}}{\sum_i f_{n_i}}$$

- Support and confidence must be computed during prediction.

23

Kardi Teknomo, PhD

Prediction

- The result of prediction would be $P[n_j] = f_{n_i n_j}$ where $n_i \in L, n_j \notin L$ and $n_i n_j \in \mathbf{F}$.
- We can then sort the prediction P based on the flow and set a threshold to reduce the items.
- The assumption is there is no repetition on the items that has been utilized.
  - All items are unique.
  - To make this unique assumption true, we add constraint that $n_j \notin L$.

24

Example

- Order does not matter
- Transaction Data

Transaction ID	Items from the customers who bought more than 1 items
1	Apple, Banana, Cherry, Durian
2	Apple, Durian
3	Banana, Durian
4	Durian, Banana, Cherry
5	Banana, Durian
6	Apple, Banana
7	Apple, Cherry, Durian

- Create a recommendation system based on ideal flow network.
- What should the system suggest for a new customer who previously have bought Cherry?

Answer 1/10

- Because order of items in a transaction does not matter, we shall create a recommendation system based on unordered data association in ideal flow network.
  - Because we have only four unique items, the IFN would have only four nodes, one for each item.
  - The trajectory data of link permutation would be used for unordered data association in IFN = Complete Graph.
  - Based on trajectory readings, we will create the network structure and the capacity matrix.
  - Finally, we compute the stochastic matrix and IFN

Answer 2/10

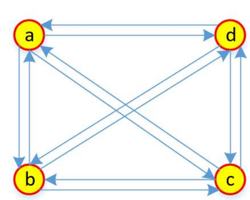
- Initially, our capacity matrix would be empty, only consist of the items on the rows and columns.

Items	Apple	Banana	Cherry	Durian
Apple				
Banana				
Cherry				
Durian				

Answer 3/10

- For the first transaction, based on the first letter of the items we have $a \rightarrow b \rightarrow c \rightarrow d$
- We read this node sequence as link permutation as follow
  - $a \rightarrow b,\ b \rightarrow a$
  - $a \rightarrow c,\ c \rightarrow a$
  - $a \rightarrow d,\ d \rightarrow a$
  - $b \rightarrow c,\ c \rightarrow b$
  - $b \rightarrow d,\ d \rightarrow b$
  - $c \rightarrow d,\ d \rightarrow c$

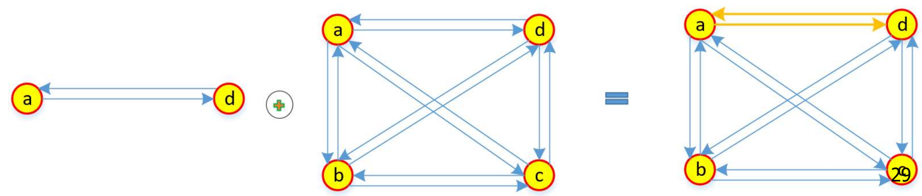
Items	Apple	Banana	Cherry	Durian
Apple		1	1	1
Banana	1		1	1
Cherry	1	1		1
Durian	1	1	1	



Answer 4/10

- Based on the second transaction data, we have $a \rightarrow d$ and the link permutation would include $d \rightarrow a$. We update the capacity matrix to become

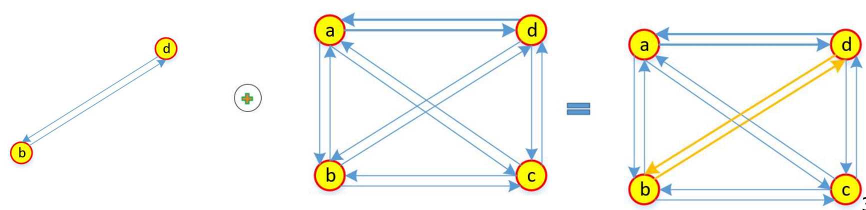
Items	Apple	Banana	Cherry	Durian
Apple		1	1	2
Banana	1		1	1
Cherry	1	1		1
Durian	2	1	1	



Answer 5/10

- Based on the third customer data, we have $b \rightarrow d$ and $d \rightarrow b$. The capacity matrix is updated into

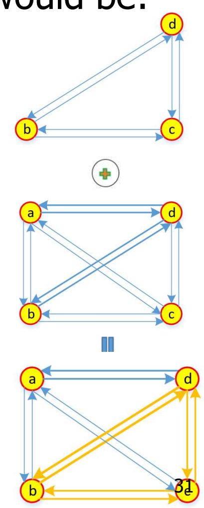
Items	Apple	Banana	Cherry	Durian
Apple		1	1	2
Banana	1		1	2
Cherry	1	1		1
Durian	2	2	1	



Kardi Teknomo, PhD

Answer 6/10

- Based on the fourth transaction data, we have $d \to b \to c$ and the link permutation would be:
  - $d \to b,\ b \to d$
  - $d \to c,\ c \to d$
  - $b \to c,\ c \to b$
- The capacity matrix is updated into



Items	Apple	Banana	Cherry	Durian
Apple		1	1	2
Banana	1		2	3
Cherry	1	2		2
Durian	2	3	2	

31

Kardi Teknomo, PhD

Answer 7/10

- Because that is our last data, the last capacity matrix would be the final matrix that we can use to compute the IFN. We have the capacity matrix

$$\mathbf{C} = \begin{array}{c} \nearrow \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} - & 2 & 2 & 3 \\ 2 & - & 2 & 4 \\ 2 & 2 & - & 3 \\ 3 & 4 & 3 & - \end{bmatrix} \end{array}$$

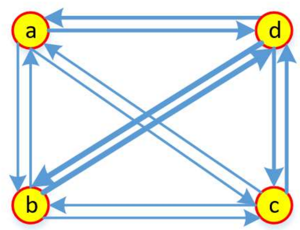
- The capacity matrix is irreducible and symmetric.

32

Answer 8/10

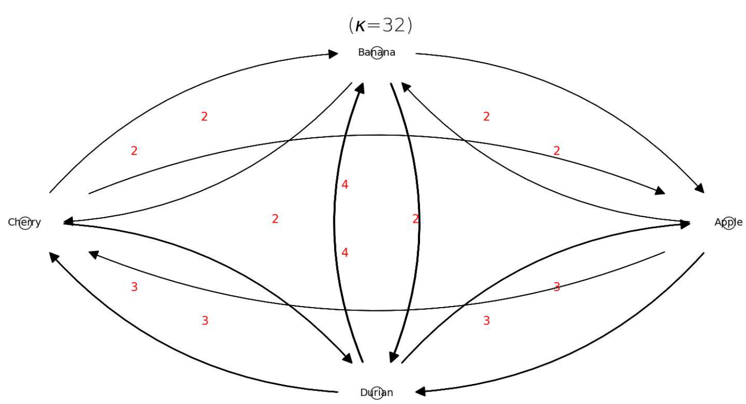
- The IFN theory stated that when the capacity matrix is irreducible and premagic, then the ideal flow matrix equivalent to the capacity matrix up to the scale. Thus, the ideal flow matrix would be

$$\mathbf{F} = \begin{array}{c|cccc|c} \nearrow & a & b & c & d & \Sigma \\ \hline a & - & 2 & 2 & 3 & 7 \\ b & 2 & - & 2 & 4 & 8 \\ c & 2 & 2 & - & 3 & 7 \\ d & 3 & 4 & 3 & - & 10 \\ \hline \Sigma & 7 & 8 & 7 & 10 & 32 \end{array}$$



Answer 9/10

$$\mathbf{F} = \begin{array}{c|cccc|c} \nearrow & a & b & c & d & \Sigma \\ \hline a & - & 2 & 2 & 3 & 7 \\ b & 2 & - & 2 & 4 & 8 \\ c & 2 & 2 & - & 3 & 7 \\ d & 3 & 4 & 3 & - & 10 \\ \hline \Sigma & 7 & 8 & 7 & 10 & 32 \end{array}$$



Kardi Teknomo, PhD

Answer 10/10

- **Prediction**: For the new customers who bought Cherry, the system would recommend Durian because among all items of the third row, node d get the highest flow ($f_{cd} = 3$). This would give us 42.85% matched (= 3/7).

```
Documents/Kardi/Personal/Tutorial/NetworkScience/IdealFlow/
Software/Python/Data Science/Experiment/Exp17')
trained network:
 {'Apple': {'Banana': 2, 'Cherry': 2, 'Durian': 3}, 'Banana':
{'Apple': 2, 'Cherry': 2, 'Durian': 4}, 'Cherry': {'Apple': 2,
'Banana': 2, 'Durian': 3}, 'Durian': {'Apple': 3, 'Banana': 4,
'Cherry': 3}}

is trained network an IFN =  True
node flow =  {'Apple': 7, 'Banana': 8, 'Cherry': 7, 'Durian':
10}
total flow =  32
Cherry to  Apple  flow =  2
Cherry to  Banana  flow =  2
Cherry to  Durian  flow =  3
prediction =  {'Durian': 3, 'Apple': 2, 'Banana': 2}
```

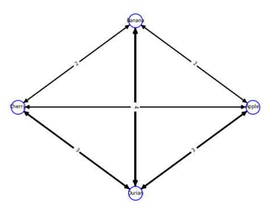
35

Kardi Teknomo, PhD

Rank

- **Rank the fruits based on the popularity**.
- The sum of rows or columns in IFN represents the strength of each item in term of its how often the items is bought among all customers. Sorting the order of the strength, we have
  - Rank #1: Durian (31.25% = 10/32)
  - Rank #2: Banana (25% = 8/32)
  - Rank #3: Apple and Cherry (21.88% = 7/32)

- $$\mathbf{F} = \begin{array}{c|cccc|c} \nearrow & a & b & c & d & \Sigma \\ \hline a & - & 2 & 2 & 3 & 7 \\ b & 2 & - & 2 & 4 & 8 \\ c & 2 & 2 & - & 3 & 7 \\ d & 3 & 4 & 3 & - & 10 \\ \hline \Sigma & 7 & 8 & 7 & 10 & 32 \end{array}$$



36

Relative Strength

Kardi Teknomo, PhD

- The **relative strength** of Apple is 7 and Banana is 8. Then we can say that Banana is 1.143 times (=8/7) more often bought than Apple.

- $$\mathbf{F} = \begin{array}{c|cccc|c} \nearrow & a & b & c & d & \Sigma \\ \hline a & - & 2 & 2 & 3 & 7 \\ b & 2 & - & 2 & 4 & 8 \\ c & 2 & 2 & - & 3 & 7 \\ d & 3 & 4 & 3 & - & 10 \\ \hline \Sigma & 7 & 8 & 7 & 10 & 32 \end{array}$$

Prediction

Kardi Teknomo, PhD

- A new customer bought Banana and Cherry. What would be the recommendation system suggest for the other items to be purchase?

- $$\mathbf{F} = \begin{array}{c|cccc|c} \nearrow & a & b & c & d & \Sigma \\ \hline a & - & 2 & 2 & 3 & 7 \\ b & 2 & - & 2 & 4 & 8 \\ c & 2 & 2 & - & 3 & 7 \\ d & 3 & 4 & 3 & - & 10 \\ \hline \Sigma & 7 & 8 & 7 & 10 & 32 \end{array}$$

Prediction

- Print out of the IFN Association Program

```
trained network:
 {'Apple': {'Banana': 2, 'Cherry': 2, 'Durian': 3}, 'Banana':
{'Apple': 2, 'Cherry': 2, 'Durian': 4}, 'Cherry': {'Apple': 2,
'Banana': 2, 'Durian': 3}, 'Durian': {'Apple': 3, 'Banana': 4,
'Cherry': 3}}

is trained network an IFN =  True
node flow =  {'Apple': 7, 'Banana': 8, 'Cherry': 7, 'Durian':
10}
total flow =  32
Banana to  Apple  flow =  2
Banana to  Durian  flow =  4
Cherry to  Apple  flow =  2
Cherry to  Durian  flow =  3
prediction =  {'Durian': 7, 'Apple': 4}
```


Conclusion

- We have extended the IFN application into recommendation system of unordered itemsets
- Based on the IFN recommendation system, we can
  - Do online **prediction** based on the latest purchase of the customer
  - **Rank the popularity** of the most frequent itemsets
  - Obtain the **relative strength** of each itemset