Extensions of Association Analysis

Dr. Meng Qu
Rutgers University

THE STATE UNIVERSITY OF NEW JERSEY
RUTGERS

Association Analysis: Advanced Concepts

Extensions of Association Analysis to Continuous and Categorical Attributes and Multi-level Rules

Continuous and Categorical Attributes

How to apply association analysis to non-asymmetric binary variables?

Gender	···	Age	Annual Income	No of hours spent online per week	No of email accounts	Privacy Concern
Female	···	26	90K	20	4	Yes
Male	···	51	135K	10	2	No
Male	···	29	80K	10	3	Yes
Female	···	45	120K	15	3	Yes
Female	···	31	95K	20	5	Yes
Male	···	25	55K	25	5	Yes
Male	···	37	100K	10	1	No
Male	···	41	65K	8	2	No
Female	···	26	85K	12	1	No
···	···	···	···	···	···	···

Example of Association Rule:

$\{\text{Gender=Male, Age} \in [21,30)\} \rightarrow \{\text{No of hours online} \geq 10\}$

Handling Categorical Attributes

- Example: Internet Usage Data

Gender	Level of Education	State	Computer at Home	Online Auction	Chat Online	Online Banking	Privacy Concerns
Female	Graduate	Illinois	Yes	Yes	Daily	Yes	Yes
Male	College	California	No	No	Never	No	No
Male	Graduate	Michigan	Yes	Yes	Monthly	Yes	Yes
Female	College	Virginia	No	Yes	Never	Yes	Yes
Female	Graduate	California	Yes	No	Never	No	Yes
Male	College	Minnesota	Yes	Yes	Weekly	Yes	Yes
Male	College	Alaska	Yes	Yes	Daily	Yes	No
Male	High School	Oregon	Yes	No	Never	No	No
Female	Graduate	Texas	No	No	Monthly	No	No
...	...	...	...	...	...	...	...

{Level of Education=Graduate, Online Banking=Yes}
$\rightarrow$ {Privacy Concerns = Yes}

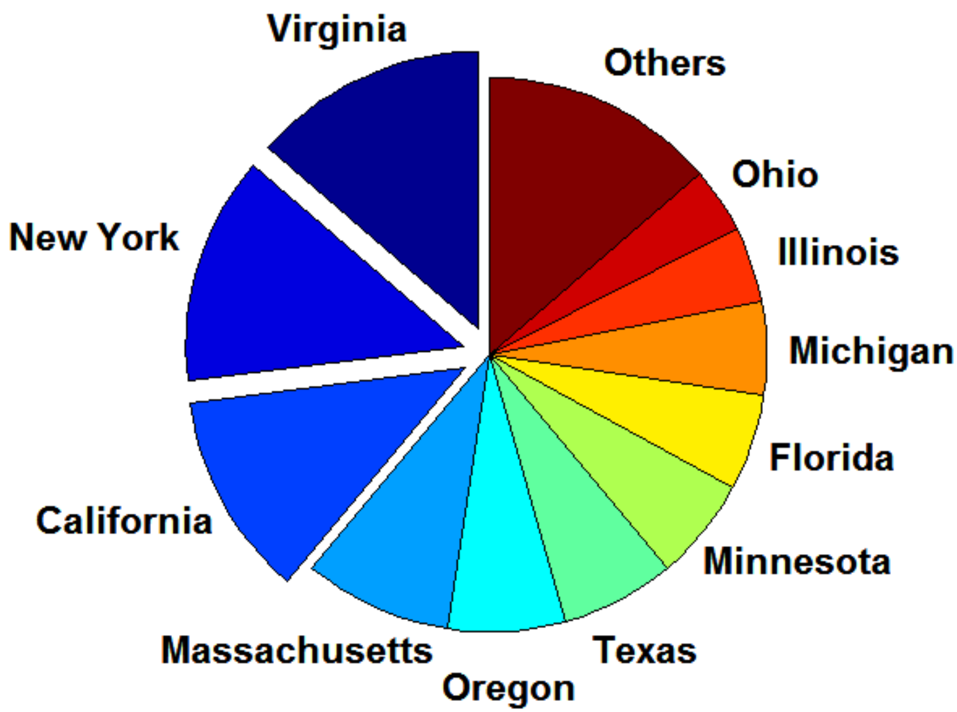
Handling Categorical Attributes

- Introduce a new "item" for each distinct attribute-value pair

Male	Female	Education = Graduate	Education = College	Education = High School	...	Privacy = Yes	Privacy = No
0	1	1	0	0	...	1	0
1	0	0	1	0	...	0	1
1	0	1	0	0	...	1	0
0	1	0	1	0	...	1	0
0	1	1	0	0	...	1	0
1	0	0	1	0	...	1	0
1	0	0	0	0	...	0	1
1	0	0	0	1	...	0	1
0	1	1	0	0	...	0	1
...	...	...	...	...	...	...	...

Handling Categorical Attributes

- Some attributes can have many possible values
  - Many of their attribute values have very low support
    - Potential solution: Aggregate the low-support attribute values



Handling Categorical Attributes

- Distribution of attribute values can be highly skewed
  - Example: 85% of survey participants own a computer at home
    - Most records have Computer at home = Yes
    - Computation becomes expensive; many frequent itemsets involving the binary item (Computer at home = Yes)
    - Potential solution:
      - discard the highly frequent items
      - Use alternative measures such as h-confidence
- Computational Complexity
  - Binarizing the data increases the number of items
  - But the width of the “transactions” remain the same as the number of original (non-binarized) attributes
  - Produce more frequent itemsets but maximum size of frequent itemset is limited to the number of original attributes

Handling Continuous Attributes

- Different methods:
  - Discretization-based
  - Statistics-based
  - Non-discretization based
    - minApriori

- Different kinds of rules can be produced:
  - {Age $\in$[21,30), No of hours online $\in$[10,20)} $\rightarrow$ {Chat Online =Yes}
  - {Age $\in$[21,30), Chat Online = Yes} $\rightarrow$ No of hours online: μ=14, σ=4

Discretization-based Methods

Gender	⋯	Age	Annual Income	No of hours spent online per week	No of email accounts	Privacy Concern
Female	⋯	26	90K	20	4	Yes
Male	⋯	51	135K	10	2	No
Male	⋯	29	80K	10	3	Yes
Female	⋯	45	120K	15	3	Yes
Female	⋯	31	95K	20	5	Yes
Male	⋯	25	55K	25	5	Yes
Male	⋯	37	100K	10	1	No
Male	⋯	41	65K	8	2	No
Female	⋯	26	85K	12	1	No
⋯	⋯	⋯	⋯	⋯	⋯	⋯

Male	Female	⋯ ⋯	Age < 13	Age $\in [13, 21)$	Age $\in [21, 30)$	⋯ ⋯	Privacy $=$ Yes	Privacy $=$ No
0	1	⋯	0	0	1	⋯	1	0
1	0	⋯	0	0	0	⋯	0	1
1	0	⋯	0	0	1	⋯	1	0
0	1	⋯	0	0	0	⋯	1	0
0	1	⋯	0	0	0	⋯	1	0
1	0	⋯	0	0	1	⋯	1	0
1	0	⋯	0	0	0	⋯	0	1
1	0	⋯	0	0	0	⋯	0	1
0	1	⋯	0	0	1	⋯	0	1
⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯

Discretization-based Methods

- Unsupervised:
  - Equal-width binning
  - Equal-depth binning
  - Cluster-based

- Supervised discretization

Continuous attribute, v

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9
Chat Online = Yes	0	0	20	10	20	0	0	0	0
Chat Online = No	150	100	0	0	0	100	100	150	100
	bin_1		bin_2			bin_3			

Discretization Issues

- Interval too wide (e.g., Bin size= 30)
  - May merge several disparate patterns
    - Patterns A and B are merged together
  - May lose some of the interesting patterns
    - Pattern C may not have enough confidence

- Interval too narrow (e.g., Bin size = 2)
  - We may lose some patterns because of their lack of support.

- Potential solution: use all possible intervals
  - Start with narrow intervals
  - Consider all possible mergings of adjacent intervals

Discretization Issues

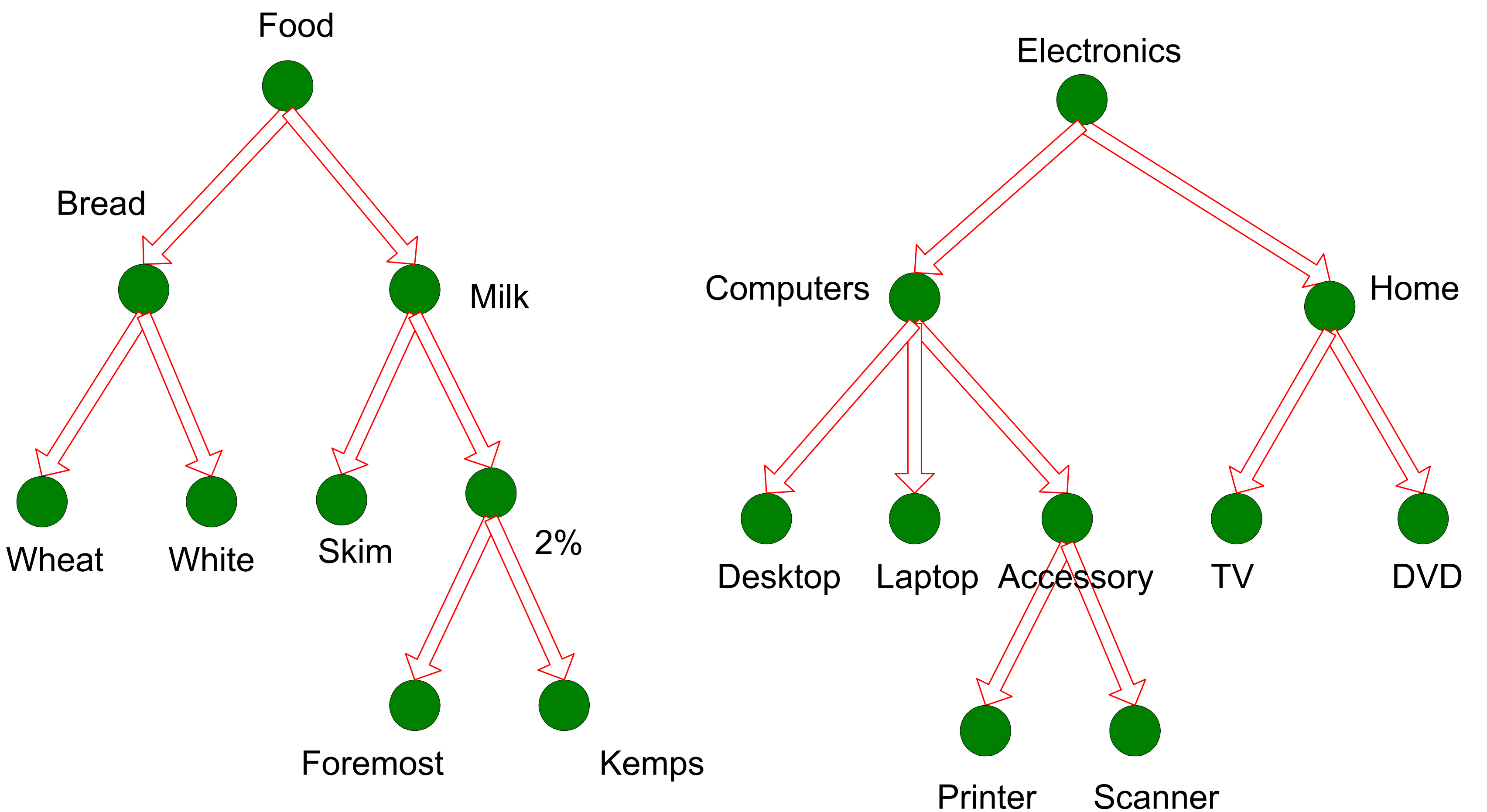
- Redundant rules

  R1: {Age $\in$[18,20), Age $\in$[10,12)} $\rightarrow$ {Chat Online=Yes}

  R2: {Age $\in$[18,23), Age $\in$[10,20)} $\rightarrow$ {Chat Online=Yes}

  – If both rules have the same support and confidence, prune the more specific rule (R1)

Concept Hierarchies



Multi-level Association Rules

- Why should we incorporate concept hierarchy?
  - Rules at lower levels may not have enough support to appear in any frequent itemsets
  - Rules at lower levels of the hierarchy are overly specific
    - e.g., skim milk $\rightarrow$ white bread, 2% milk $\rightarrow$ wheat bread, skim milk $\rightarrow$ wheat bread, etc.

    are indicative of association between milk and bread

Multi-level Association Rules

- Approach 1:
  - Extend current association rule formulation by augmenting each transaction with higher level items

Original Transaction: {skim milk, wheat bread}

Augmented Transaction:
{skim milk, wheat bread, milk, bread, food}

- Issues:
  - Items that reside at higher levels have much higher support counts
    - if support threshold is low, too many frequent patterns involving items from the higher levels
  - Increased dimensionality of the data

Multi-level Association Rules

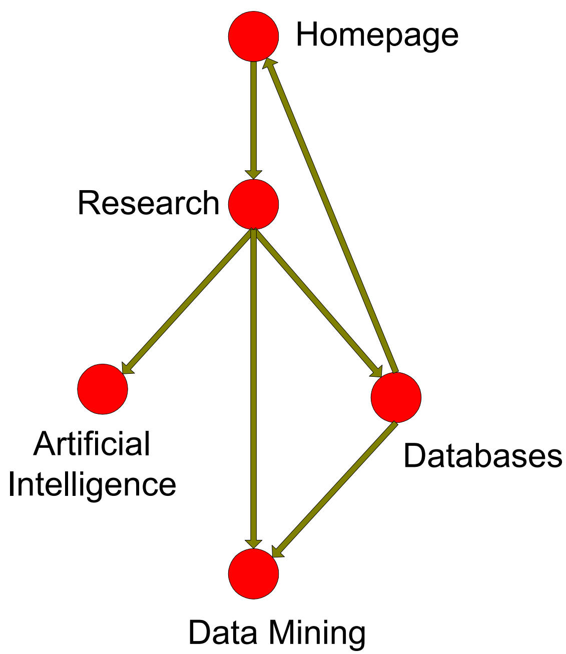
- Approach 2:
  - Generate frequent patterns at highest level first
  - Then, generate frequent patterns at the next highest level, and so on
- Issues:
  - I/O requirements will increase dramatically because we need to perform more passes over the data
  - May miss some potentially interesting cross-level association patterns

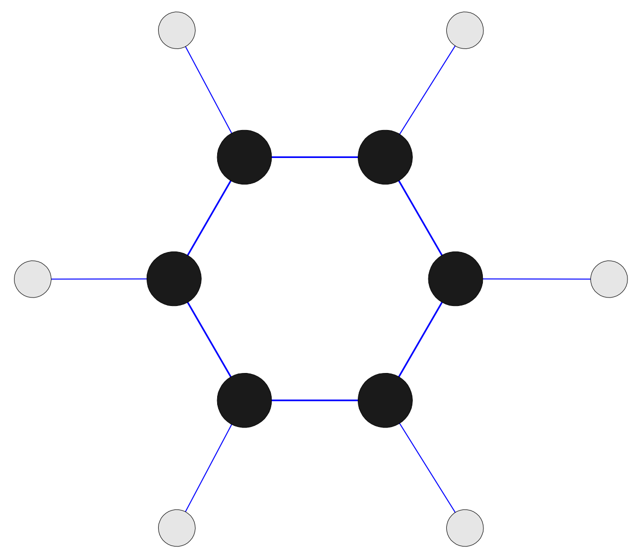
Association Analysis: Advanced Concepts

Subgraph Mining

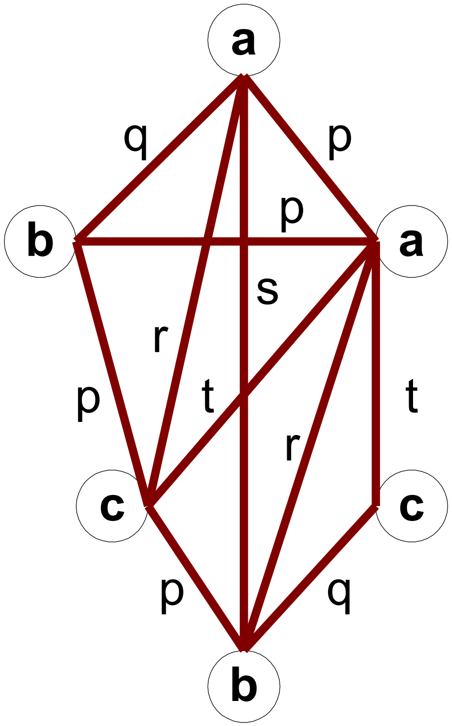
Frequent Subgraph Mining

- Extends association analysis to finding frequent subgraphs
- Useful for Web Mining, computational chemistry, bioinformatics, spatial data sets, etc

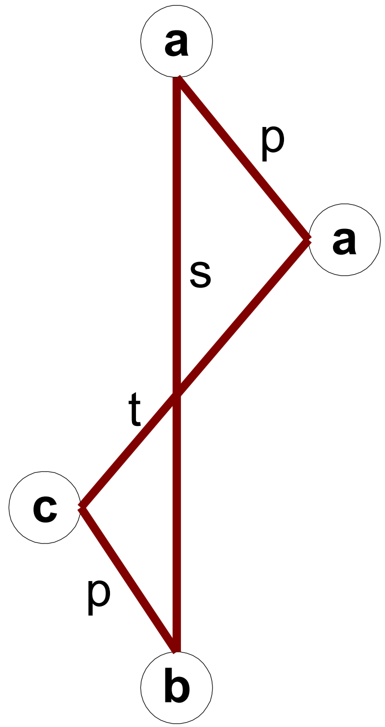




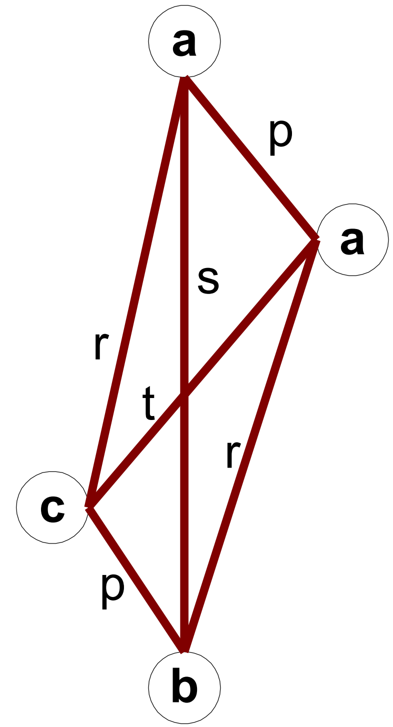
Graph Definitions



(a) Labeled Graph



(b) Subgraph

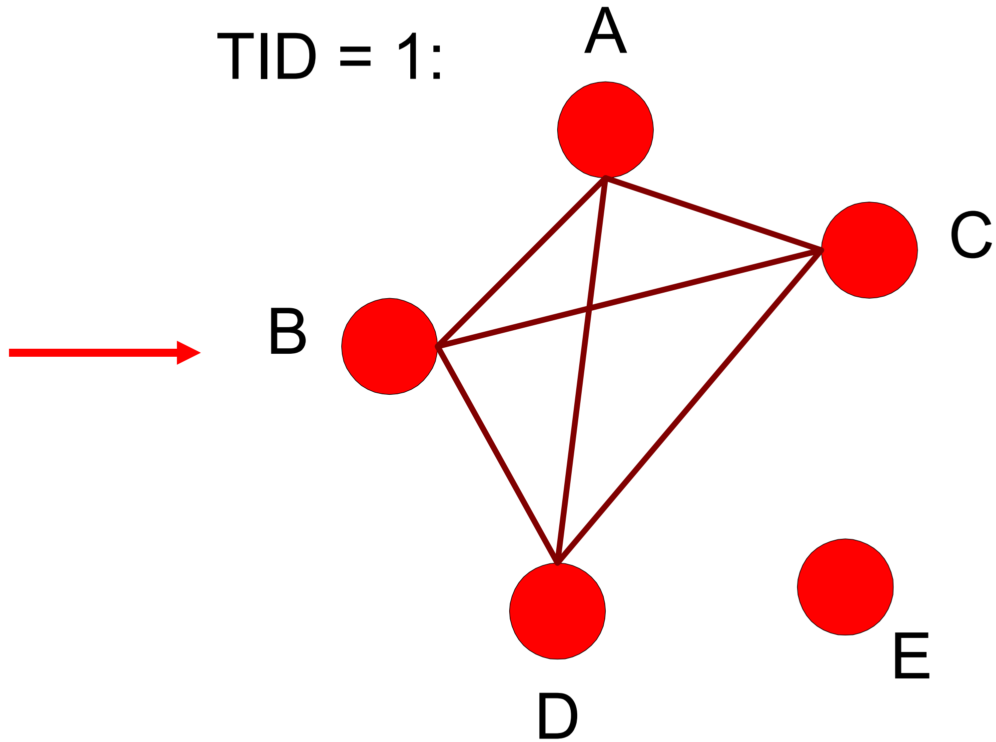


(c) Induced Subgraph

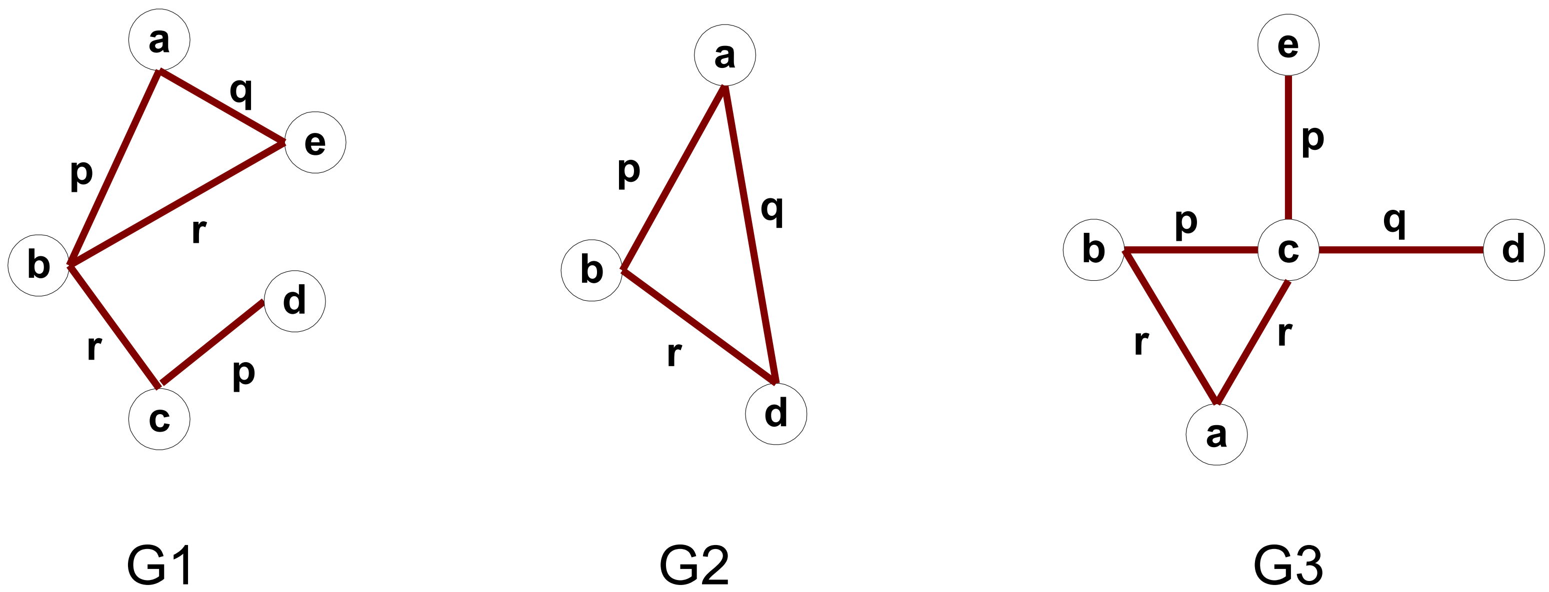
Representing Transactions as Graphs

- Each transaction is a clique of items

Transaction Id	Items
1	{A,B,C,D}
2	{A,B,E}
3	{B,C}
4	{A,B,D,E}
5	{B,C,D}



Representing Graphs as Transactions



	(a,b,p)	(a,b,q)	(a,b,r)	(b,c,p)	(b,c,q)	(b,c,r)	…	(d,e,r)
G1	1	0	0	0	0	1	…	0
G2	1	0	0	0	0	0	…	0
G3	0	0	1	1	0	0	…	0
G3	…	…	…	…	…	…	…	…

Challenges

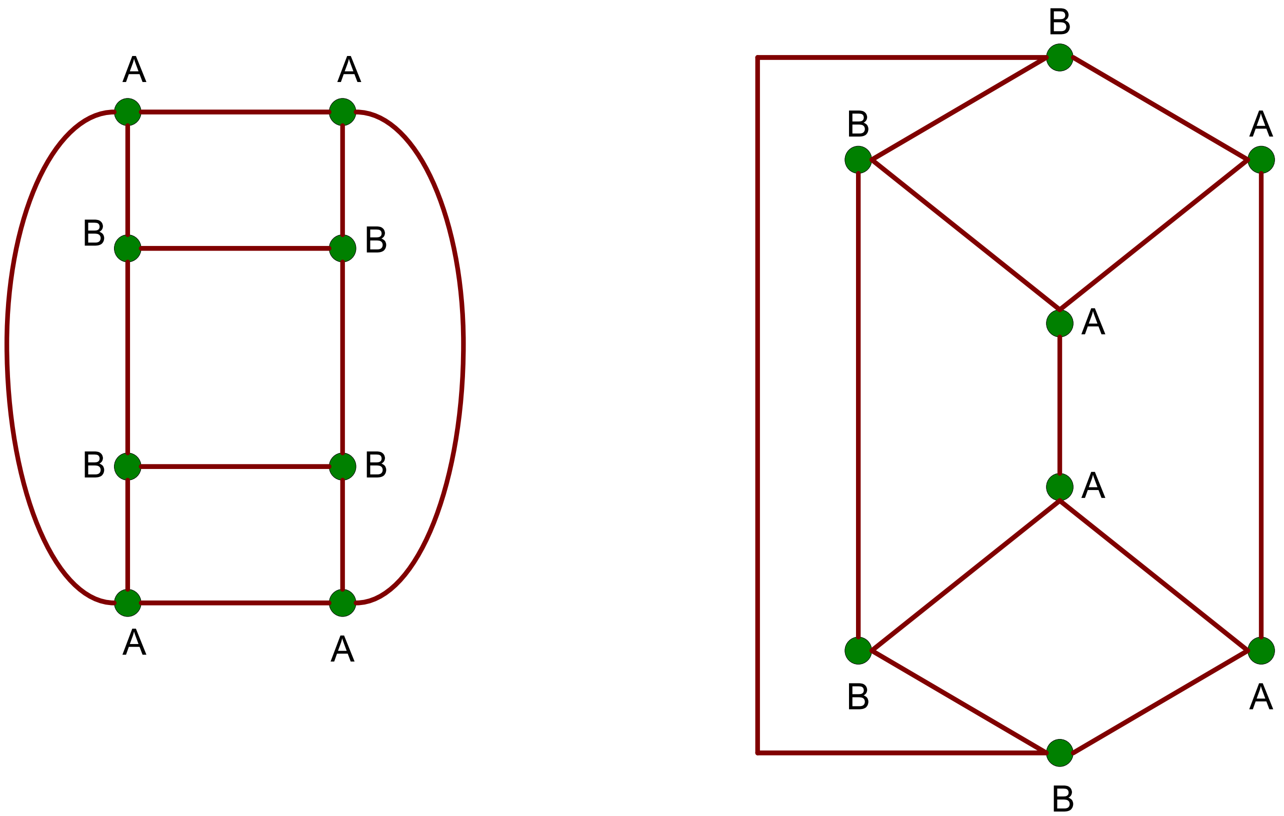
- Node may contain duplicate labels
- Support and confidence
  - How to define them?
- Additional constraints imposed by pattern structure
  - Support and confidence are not the only constraints
  - Assumption: frequent subgraphs must be connected
- Apriori-like approach:
  - Use frequent k-subgraphs to generate frequent (k+1) subgraphs
    - What is k?

Challenges…

- Support:
  - number of graphs that contain a particular subgraph
- Apriori principle still holds
- Level-wise (Apriori-like) approach:
  - Vertex growing:
    - k is the number of vertices
  - Edge growing:
    - k is the number of edges

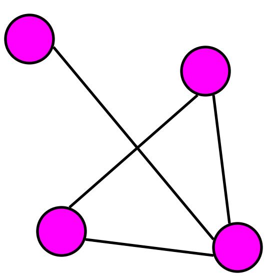
Graph Isomorphism

- A graph is isomorphic if it is topologically equivalent to another graph



Graph Isomorphism

- Use canonical labeling to handle isomorphism
  - Map each graph into an ordered string representation (known as its code) such that two isomorphic graphs will be mapped to the same canonical encoding
  - Example:
    - Lexicographically largest adjacency matrix



$$\rightarrow \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

String: 011011 **Canonical: 111100**