DATA MINING AND WAREHOUSING

Graph Mining

DHANYAJA N
Assistant Professor
STM Kannur

Contents

- Graph Mining: Introduction
- Application
- Methods for finding frequent sub graph
- Apriori based approach

Graph Mining

- *Graph Mining* is the set of tools and techniques used to

(a) analyze the properties of real-world graphs

(b) predict how the structure and properties of a given graph might affect some application

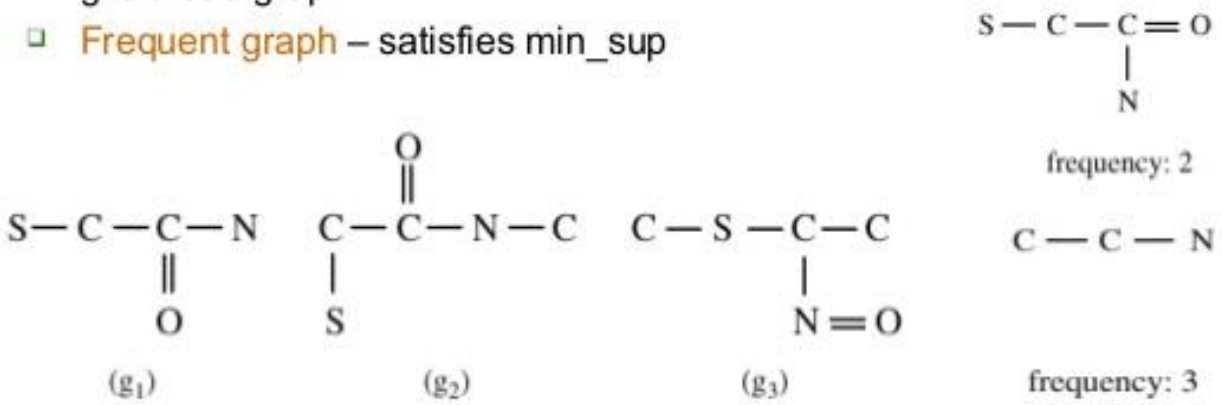
(c) develop models that can generate realistic graphs that match the patterns found in real-world graphs of interest.

Graph Mining

- Graphs
  - Model sophisticated structures and their interactions
    - Chemical Informatics
    - Bioinformatics
    - Computer Vision
    - Video Indexing
    - Text Retrieval
    - Web Analysis
    - Social Networks
  - Mining frequent sub-graph patterns
    - Characterization, Discrimination, Classification and Cluster Analysis, building graph indices and similarity search

Mining Frequent subgraph

- Graph g
  - Vertex Set – V(g)
  - Edge set – E(g)
  - Label function maps a vertex / edge to a label
  - Graph g is a sub-graph of another graph g' if there exists a graph iso-morphism from g to g'
  - Support(g) or frequency(g) – number of graphs in D = {G_1, G_2,..G_n} where g is a sub-graph
  - Frequent graph – satisfies min_sup



Methods for Mining Frequent Sub graphs

- Apriori-based Approach
- ***Apriori-based algorithms*** for frequent substructure mining include AGM, FSG, and a path-join method

- AGM shares similar characteristics with Apriori-based item-set mining.
-
- FSG and the path-join method explore edges and connections in an Apriori-based fashion

Apriori Based Approach

Algorithm: AprioriGraph. Apriori-based frequent substructure mining.

Input:

- D, a graph data set;
- min_sup, the minimum support threshold.

Output:

- S_k, the frequent substructure set.

Method:

$S_1 \leftarrow$ frequent single-elements in the data set;
Call AprioriGraph(D, min_sup, S_1);

procedure AprioriGraph(D, min_sup, S_k)

(1) $S_{k+1} \leftarrow \varnothing$;
(2) **for each** frequent $g_i \in S_k$ **do**
(3) **for each** frequent $g_j \in S_k$ **do**
(4) **for each** size $(k+1)$ graph g formed by the merge of g_i and g_j **do**
(5) **if** g is frequent in D and $g \notin S_{k+1}$ **then**
(6) insert g into S_{k+1};
(7) **if** $s_{k+1} \neq \varnothing$ **then**
(8) *AprioriGraph*(D, min_sup, S_{k+1});
(9) **return**;

Start with graph of small size – generate candidates with extra vertex/edge or path

AprioriGraph

- Level wise mining method
- Size of new substructures is increased by 1
- Generated by joining two similar but slightly different frequent sub-graphs
- Frequency is then checked

Candidate generation in graphs is complex

Apriori Based Approach

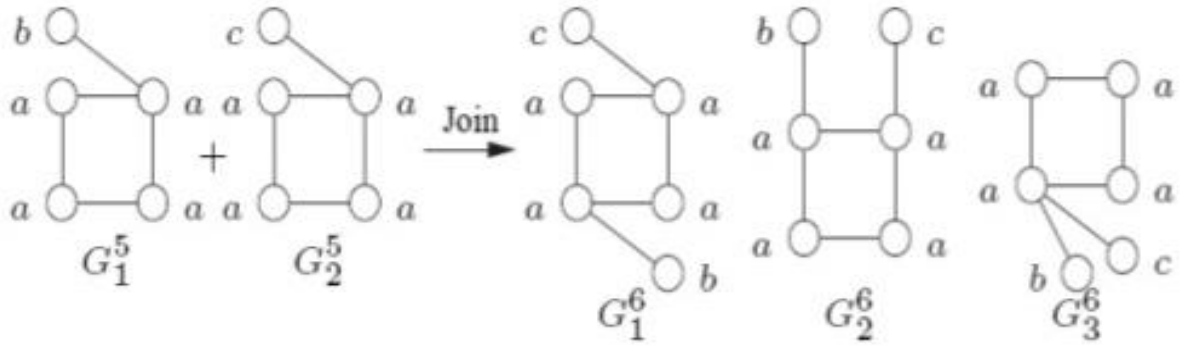
AGM (Apriori-based Graph Mining)

- Vertex based candidate generation – increases sub structure size by one vertex at each step
- Two frequent k size graphs are joined only if they have the same (k-1) subgraph (Size – number of vertices)
- New candidate has (k-1) sized component and the additional two vertices
  - Two different sub-structures can be formed

Apriori Based Approach

FSG (Frequent Sub-graph mining)

- Edge-based Candidate generation – increases by one-edge at a time
- Two size k patterns are merged iff they share the same subgraph having k-1 edges (core)
- New candidate – has core and the two additional edges



Apriori Based Approach

Edge disjoint path method

- Classify graphs by number of disjoint paths they have
- Two paths are edge-disjoint if they do not share any common edge
- A substructure pattern with k+1 disjoint paths is generated by joining sub-structures with k disjoint paths

Disadvantage of Apriori Approaches

- Overhead when joining two sub-structures
- Uses BFS strategy : level-wise candidate generation
  - To check whether a k+1 graph is frequent – it must check all of its size-k sub graphs
  - May consume more memory

Thank You !!!

DHANYAJA N
Assistant Professor
STM Kannur