Data Mining, Lecture 13

Mining Graph Data

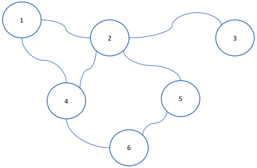
S. Nõmm

[1] Department of Software Science, Tallinn University of Technology

02.12.2021

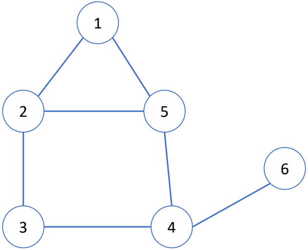
Introduction

- The structure may be more important compared to content.
- Applications: physics, biology, social studies.



Non oriented graph representation

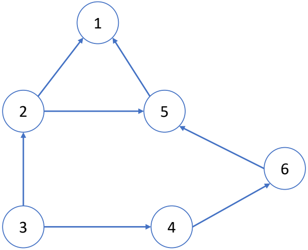
Described by the list or by adjacency matrix



	1	2	3	4	5	6
1	0	1	0	0	1	0
2	1	0	1	0	1	0
3	0	1	0	1	0	0
4	0	1	1	0	1	1
5	1	1	0	1	0	0
6	0	0	0	1	0	0

Oriented graph description

Adjacent matrix is non symmetric.



Origin (who)

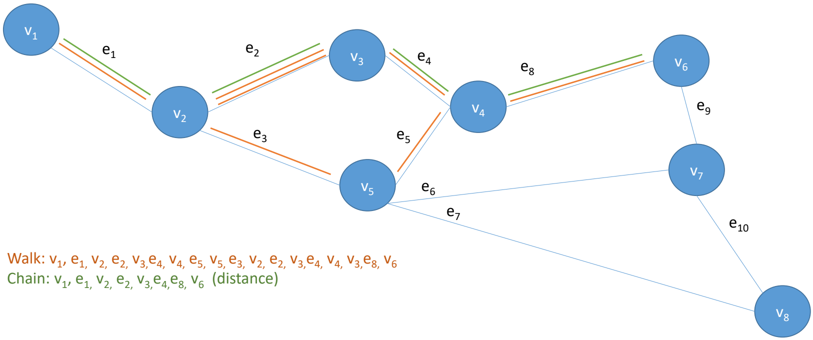
Terminal

	1	2	3	4	5	6
1	0	1	0	0	1	0
2	0	0	1	0	0	0
3	0	0	0	0	0	0
4	0	0	1	0	0	0
5	0	1	0	0	0	1
6	0	0	0	1	0	0

Path & Walk (chain)

- Walk in the graph G is the sequence $v_0, e_1, v_1, \dots e_l, v_l$, where v_i are nodes (vertexes) and e_i are the ages between the vertexes.
- Vertex v_0 is referred as initial vertex and v_l terminal vertex.
- Path is the walk with no repetitions.
- Vertex v_i is reachable from the vertex v_j if thehere is a walk from v_i to v_j.
- The distance between v_i and v_j is defined as the shortest path between them.

Path & Walk (chain)



Walk: v_1, e_1, v_2, e_2, v_3, e_4, v_4, e_5, v_5, e_3, v_2, e_2, v_3, e_4, v_4, v_3, e_8, v_6

Chain: v_1, e_1, v_2, e_2, v_3, e_4, e_8, v_6 (distance)

Graph database

Definition

- *Graph data base $\mathcal{D}$ is defined as the collection of different undirected graphs* $G_1 = (N_1, A_1), \ldots, G_n = (N_n, A_n)$.

- The set of nodes in ith graph is denoted by N_i and the set of edges by A_i.
- Each node $p \in N_i$ is associated with the label $l(p)$.

Matching and distance computation

- The term matching is used in two distinct contexts for graph mining.
- Pairing up nodes in a single graph with the use of edges is also referred to as matching.
- Within the frameworks of the present lecture the term *matching* is used with conjunction to graph matching, the problem is also referred as graph isomorphism.

Matching and distance computation

Definition

Two graphs $G_1 = (N_1, A_1)$ and $G_2 = (N_2, A_2)$ are said to be isomorphic if there exists a bijection f between the sets of nodes N_1 and N_2, such that following two conditions are satisfied.

1. *For each pair of corresponding nodes their labels are the same.*
2. *The edge between the nodes $p_{i,1}$ and $p_{j,1}$ exists in G_1 if and only if the edge exists between the nodes $f(p_{i,2})$ and $f(p_{i,2})$ in G_2.*

Definition

A node induced subgraph of graph $G = (N, A)$ is a graph $G_s = (N_s, A_s)$ satisfying two properties:

1. $N_s \subseteq N$.
2. $A_s = A \cap (N_s \times N_s)$.

Matching and distance computation

Definition

A query graph $G_q = (N_q, A_q)$, is said to be a subgraph isomorphism of the data graph $G = (N, A)$ if two following conditions are satisfied:

1. *For each node $p_i \in N_q$ there is exist a node $p_j \in N$ such that $l(p_i) = l(p_j)$.*
2. *The edge a_{i_1,j_1}, between the nodes $p_{i,1}$ and $p_{j,1}$, exists in G_q if and only if corresponding edge exists in G.*

Definition

A Maximal Common Subgraph between a pair of subgraphs $G_1 = (N_1, A_1)$ and $G_2 = (N_2, A_2)$ is a graph $G_0 = (N_0, A_0)$ such that it is a subgraph isomorphism for the both G_1 and G_2, whereas the power of N_0 is the maximal (of all possible).

Ullmans algorithm may be used to determine all possible subgraph isomorphisms between a query graph and a data graph.

MCG-based distances

NB! Not all of the MCG-based distances satisfy condition to be a metric.

- Unnormalized non-matching measure:
  $U(G_1, G_2) = |G_1| + |G_2| - 2 \cdot |MCS(G_1, G_2)|$.
- Union-normalized distance:
  $$U_n = (G_1, G_2) = 1 - \frac{|MCS(G_1, G_2)|}{|G_1| + |G_2| - MCS(G_1, G_2)}.$$
- Max-normalized distance:
  $$U_n^{max} = 1 - \frac{|MCS(G_1, G_2)|}{\max\{|G_1|, |G_2|\}}.$$

Edit based distances

Definition

The graph edit distance $E(G_1, G_2)$ it the minimum cost of the edit operations to be applied to G_1 in order to transform it to G_2.

item Not necessarily symmetric.

Topological descriptors

Topological descriptors convert structural graphs to multidimensional data by using quantitative measures of important structural characteristics as dimensions.

- Morgan index: equal to the number of nodes reachable from the node within a distance of k.
- Wiener index:equal to the sum of the pairwise shortest path distances between all pairs of nodes.

$$W(G) = \sum_{i,j \in G} d(i,j).$$

- Hosoya index: is equal to the number of valid pairwise node-node matchings in the graph.
- Circuit rank: is equal to the minimum number of edges that need to be removed from a graph in order to remove all cycles.

Frequent Substructure Mining in Graphs

The idea of frequent subgraph is identical to the case of association pattern mining, except that a subgraph relationship is used to count the support rather than a subset relationship.

- Let $\mathcal{G}$ - Graph Database, $minsup$ - minimum support.
- begin
- $F_1 = \{$ All Frequent singleton graphs $\}$;
- $k = 1$;
- while F_k is not empty do begin
- Generate $\mathcal{C}_{k+1}$ by joining pairs of graphs in F_k that share a subgraph of size (k 1) in common;
- Prune subgraphs from $\mathcal{C}_{k+1}$ that violate downward closure;
- Determine F_{k+1} by support counting on $(\mathcal{C}_{k+1}, \mathcal{G})$ and retaining subgraphs from $\mathcal{C}_{k+1}$ with support at least $minsup$;
- k = k + 1;
- end;
- return $(\cup_{i=1}^{k} F_i)$;
- end

Graph clustering

- The graph clustering problem partitions a database of n graphs into groups.
- Distance-based methods.
  - k-medoids
  - "community detection" (will be discussed during the next lecture)
- Frequent substructure-based methods.
  - Generic Transformational Approach
  - XProj: Direct Clustering with Frequent Subgraph Discovery

Graph Classification

- Distance-based methods.
- Frequent substructure-based methods.
  - Generic Transformational Approach
  - XRules: A Rule-Based Approach

Ullmans algorithm

- Let G_q - query graph, G - data graph, $\mathcal{M}$ currently partially matched node pairs.
- begin
- if $|\mathcal{M}| = |N_q|$ then return successful match $\mathcal{M}$
- else
- $\mathcal{C} =$ Set of all label matching node pairs from (G_q, G) not in $\mathcal{M}$
- (Optional efficiency optimization)
- for each pair $(p_{i_q}, p_i) \in \mathcal{C}$ do
- if $\mathcal{M} \cup \{(p_{i_q}, p_i)\}$ is valid partial matching
- then subgraph match $(G_q, G, \mathcal{M} \cup \{(p_{i_q}, p_i)\})$;
- end for
- end

Maximum common subgraph algorithm

- Let G_1 and G_2 - graphs, $\mathcal{M}$ currently partially matched node pairs, $\mathcal{M}_b$ currently best match .
- begin
- $\mathcal{C} =$ Set of all label matching node pairs from (G_1, G_2) not in $\mathcal{M}$
- (Optional efficiency optimization)
- for each pair $(p_{i,1}, p_{j,2}) \in \mathcal{C}$ do
- if $\mathcal{M} \cup \{(p_{i,1}, p_{j,2})\}$ is valid matching
- then $\mathcal{M}_b =$ MCG $(G_1, G_2, \mathcal{M} \cup \{(p_{i,1}, p_{j,2})\})$;
- end for
- if $(|\mathcal{M}| > |\mathcal{M}_b|)$ then return $\mathcal{M}$ else return $\mathcal{M}_b$
- end

Graph matching methods and distance computations

- Pairs of graphs that share large subgraphs in common are likely to be more similar.
- Edit distance.
- Transformation based distance computation.