Combined Structure-Weight Graph Similarity and Its Application in E-Health

Mahsa Kiani, Virendra Bhavsar, Harold Boley

Faculty of Computer Science, University of New Brunswick, Fredericton, NB, Canada

Introduction

In order to determine the similarity of rich structures, they should be represented in an expressive manner, for example, as (edge-)weighted trees, Directed Acyclic Graphs (DAGs), or graphs.

Efficient similarity algorithms are required in many applications, such as schema matching in databases, buyer-seller matching in e-Business, and health record retrieval.

Existing Methods [1] [2] only compare the **structure similarity** of the query graph with stored graphs. They cannot differentiate weighted trees or DAGs with identical structure similarity but different **weight similarity**.

Contribution

We propose a **combined structure-weight similarity algorithm** that uses structure and weight similarity values as, respectively, primary and secondary criteria to rank the retrieved graphs.

Applications

- Semantic-pragmatic information retrieval
- Social-network clustering
- Health 3.0

Representation of Data

Given query as well as stored structured data is represented as a weighted directed acyclic graph.

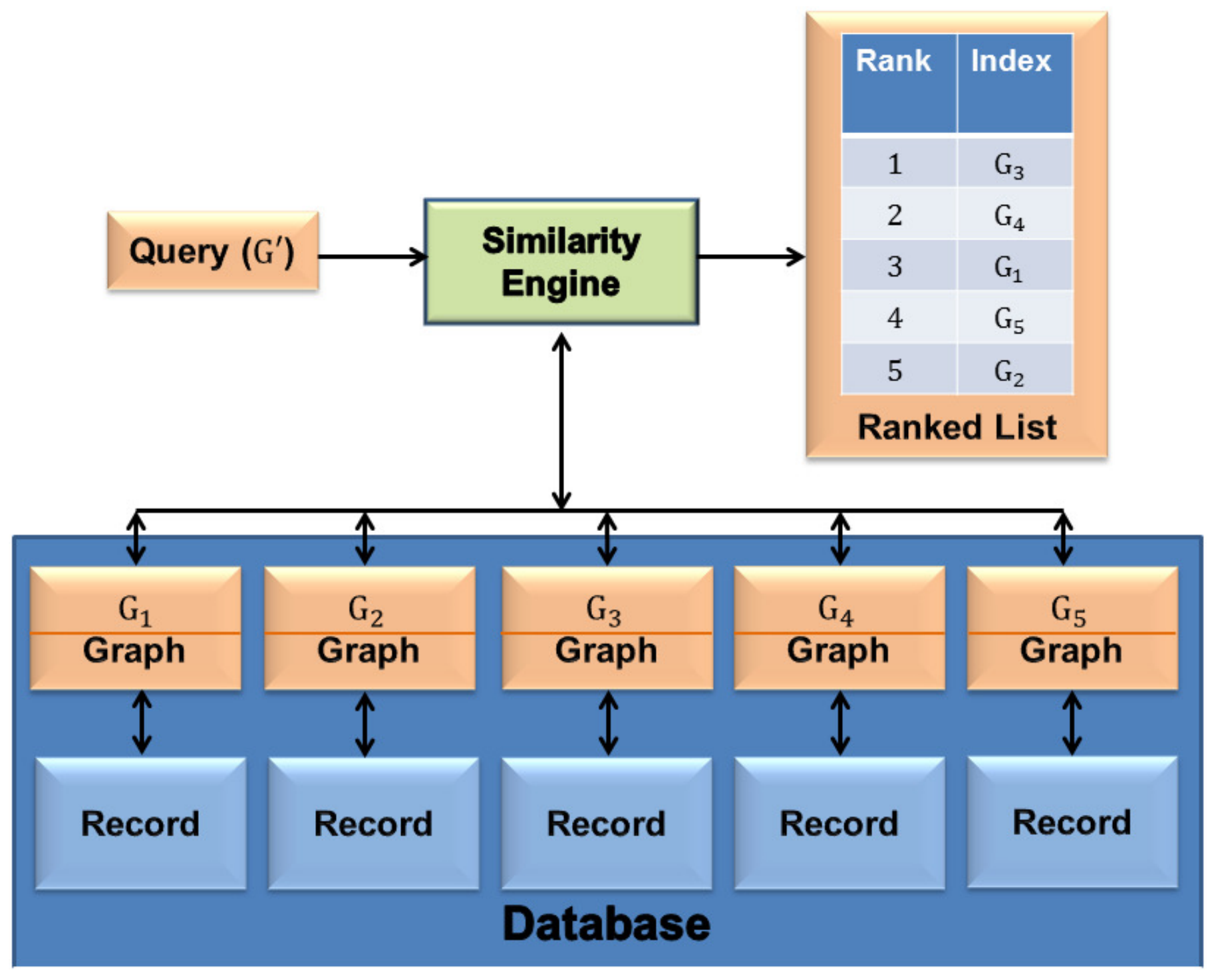
Edge weights express users' assessment regarding the relative importance of the attributes represented by edge labels.

- Labels are unique and appear in lexicographic left-to-right order.
- Weights are in the real interval [0, 1] and for each graph its edge weights are normalized.

Stored graphs and query graph are interchanged using an extension of Weighted Object Oriented RuleML [3].

System Architecture

Given a query graph, a ranked list of matching graphs (and consequently corresponding records), which are stored in a dataset, is constructed by the similarity engine.



Rank	Index
1	G_3
2	G_4
3	G_1
4	G_5
5	G_2

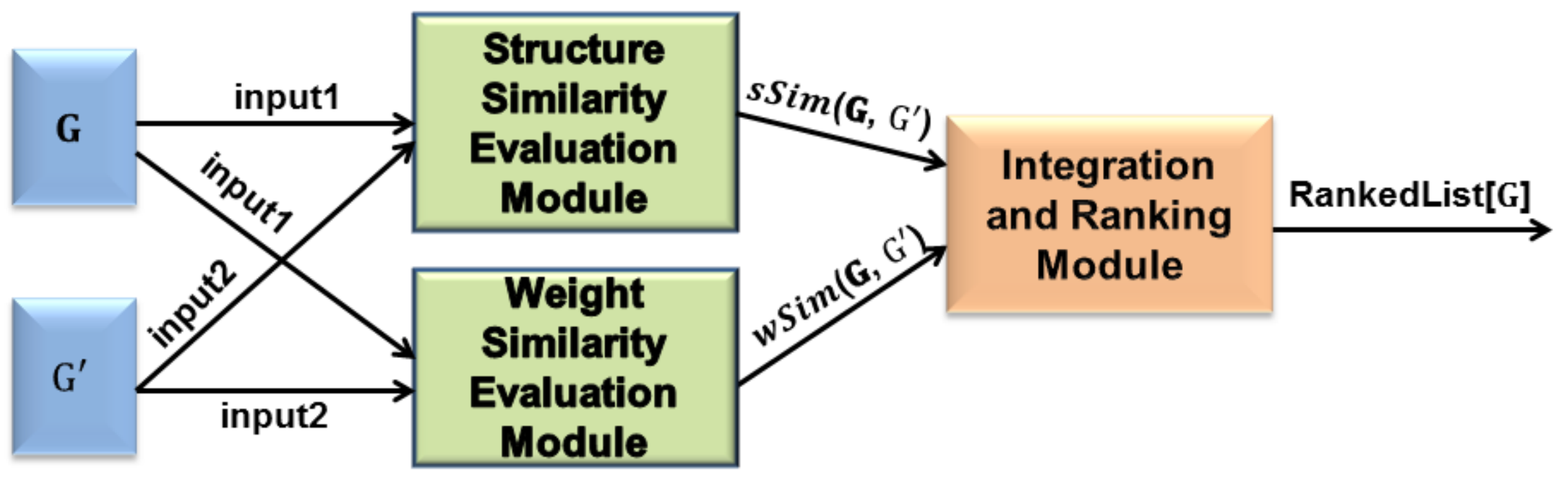
System Architecture

Natural Sciences and Engineering Research Council of Canada
Conseil de recherches en sciences naturelles et en génie du Canada
Canada

Similarity Engine Architecture

- $\mathbf{G}$ and G': A set of stored graphs and a given query graph being matched
- $sSim(\mathbf{G}, G')$: Structure similarity values
- $wSim(\mathbf{G}, G')$: Weight similarity values

The integration and ranking module ranks the graphs in $\mathbf{G}$ based on the structure similarity and weight similarity.


G
G′
input1
input2
input1
input2
Structure Similarity Evaluation Module
Weight Similarity Evaluation Module
sSim(G, G′)
wSim(G, G′)
Integration and Ranking Module
RankedList[G]

Architecture of Similarity Engine

Graph Weight Similarity

Two given graphs are traversed in a top-down (root-leaf) order to compute the edge-weight similarity of the graphs. If two edges being traversed are *corresponding edges*, their weight similarity ($WeSim_p$) is calculated based on Manhattan distance or Min/Max similarity measure. The combined edge weight similarity value is calculated using

$$Sim = \sum_{d=1}^{d_{max}} \sum_{p=1}^{m_d} WeSim_p \cdot D^{d+1}$$

- d: The depth of the source node of the edge
- d_{max}: The maximum possible depth of the source node of corresponding edges in two graphs
- p: The enumeration of the pairs of corresponding edges in depth d
- m_d: : The number of corresponding edges in depth d
- D: The global depth degradation factor ($D \leq 0.5$) which adjusts the importance of the weight similarities related to various levels of the graphs

Then, Sim is normalized by the sum of the D^{d+1} used in various iterations of the recursive weight similarity algorithm, to obtain $wSim(G_1, G')$.

Integration and Ranking Module

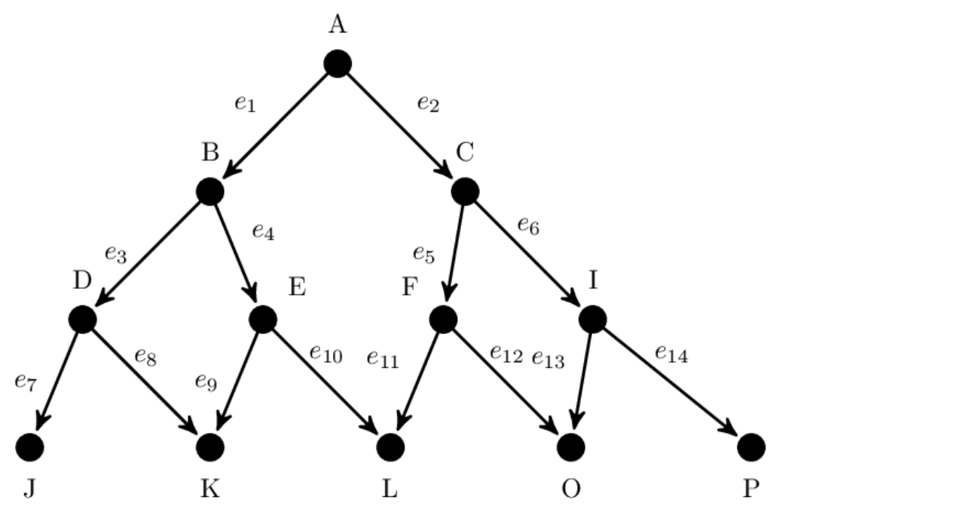
- $sSim(G_i, G')$: Structure similarity of G_i to G'
- $wSim(G_i, G')$: Weight similarity of G_i to G'
- $G_1 << G_2$: G_1 appears before G_2 in the ranked list

$G_1 << G_2$ if and only if [$sSim(G_1, G') > sSim(G_2, G')$] or [$sSim(G_1, G') = sSim(G_2, G')$ and $wSim(G_1, G') > wSim(G_2, G')$]

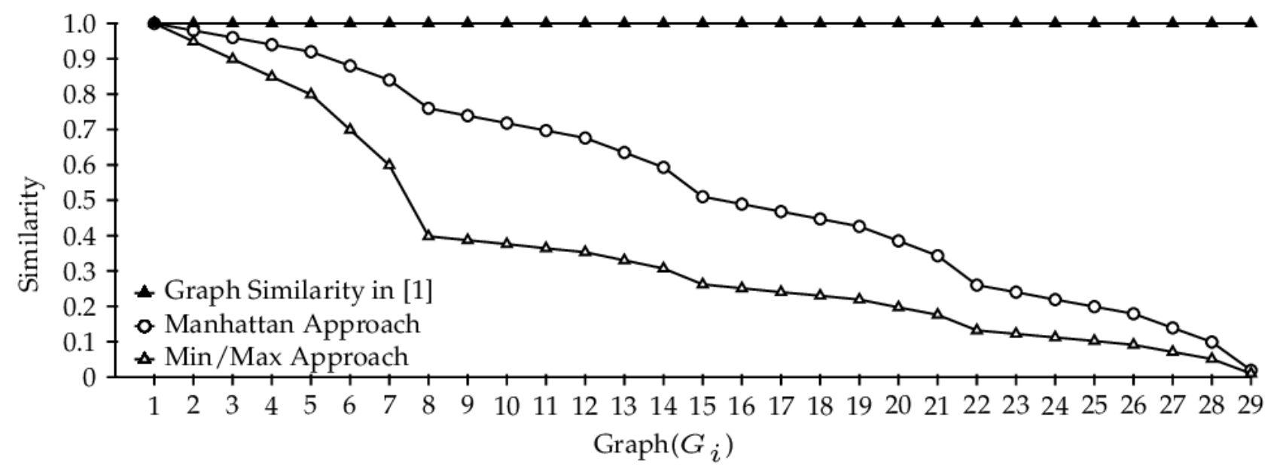
$G_1 \ll G_2$ or $G_2 \ll G_1$if [$sSim(G_1, G') = sSim(G_2, G')$ and $wSim(G_1, G') = wSim(G_2, G')$]

Computational Experiments

- Graph dataset: $\mathbf{G} = \{G_1, G_2, \ldots, G_{29}\}$
- Possible edge weights: [0.01, 0.99], [0.25, 0.25], [0.5, 0.5], [0.75, 0.25], [0.99, 0.01]
- Systematic change of weight to obtain 29 graphs

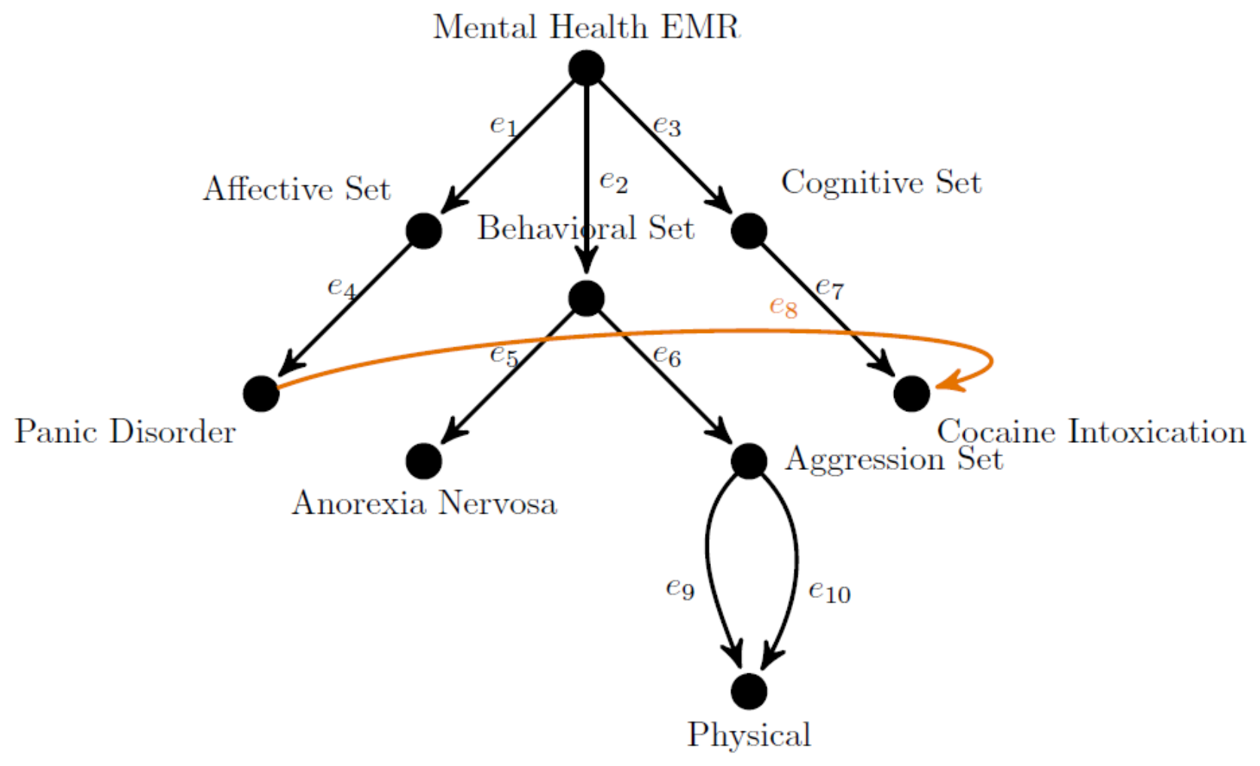


Graph Structure in Dataset



Similarity of G_1 to 29 Graphs in the Dataset

Application in E-Health



Metadata of Mental Health Records

Edge Weights

Graph	$w(e_1)$	$w(e_2)$	$w(e_3)$	$w(e_4)$	$w(e_5)$	$w(e_6)$	$w(e_7)$	$w(e_8)$	$w(e_9)$	$w(e_{10})$
G'_1	0.01	0.01	0.98	1.0	0.01	0.99	1.0	1.0	0.01	0.99
G_1	0.01	0.01	0.98	1.0	0.01	0.99	1.0	1.0	0.01	0.99
G_2	0.5	0.25	0.25	1.0	0.25	0.75	1.0	1.0	0.25	0.75
G_3	0.4	0.3	0.3	1.0	0.5	0.5	1.0	1.0	0.5	0.5
G_4	0.3	0.35	0.35	1.0	0.75	0.25	1.0	1.0	0.75	0.25

Similarity values and Ranked List

Graph	Graph	Structure Similarity	Manhattan Approach	Min/Max Approach	Rank
G'	G_1	1.0	1.0	1.0	1
G'	G_2	1.0	0.6834	0.3762	2
G'	G_3	1.0	0.6356	0.3492	3
G'	G_4	1.0	0.5878	0.3249	4

Stored metadata which have higher similarity to the treatment priorities (i.e., edge weights) of the query appear higher in the ranked results.

Conclusion

This approach leads to higher precision compared to earlier approaches that did not incorporate the similarity of edge weights.

Publication:

Kiani, M.; Bhavsar, V.C., Boley, H., "Combined Structure-Weight Graph Similarity and Its Application in E-Health", Accepted in 4th Canadian Semantic Web Symposium, Montreal, July, 2013.

References:

[1] Bhavsar, V.C., Boley, H., Yang, L., "A Weighted-Tree Similarity Algorithm for Multi-Agent Systems in E-Business Environments", Computational Intelligence, pp.584-602, 2004.

[2] Jing, J., "Similarity of Weighted Directed Acyclic Graphs", Master's Thesis, Faculty of Computer Science, University of New Brunswick, 2006.

[3] Boley, H., Grosof, B., Kifer, M., Sintek, M., Tabet, S., and Wagner, G., "Object-oriented rulemI",http://www.ruleml.org/indoo/indoo.html, 2004.

The first author would like to thank Mehrdad Kiani, M.D., who provided help in the health domain.