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

Mahsa Kiani, Virendrakumar C. Bhavsar, and Harold Boley
Faculty of Computer Science
University of New Brunswick
Fredericton, NB, Canada
{*mahsa.kiani, bhavsar, harold.boley*}*[AT]unb.ca*

***Abstract*—A combined structure-weight similarity approach for comparing directed (vertex- and edge-)labeled (edge-) weighted graphs is presented. Vertex labels (as types) and edge labels (as attributes) embody semantic information. Edge weights express assessments regarding the (percentage-)relative importance of the attributes, a kind of pragmatic information. These graphs are uniformly represented and interchanged using a weighted extension of Object Oriented RuleML. We propose semantic-pragmatic information retrieval and clustering where a combination of structure and weight similarities between a query and stored graphs is calculated. The structure and weight similarity values are used as primary and secondary criteria, respectively, to rank the retrieved graphs. The proposed weight similarity algorithm refines the ranking of retrieved graphs that have identical or nearly identical query-graph structure similarity but have different edge weights. It is shown that our approach leads to higher precision compared to earlier approaches that did not incorporate the similarity of edge weights. The proposed approach of semantic-pragmatic information retrieval and clustering can be applied, for example, in e-Learning, e-Business, social networks, and Health 3.0. In this paper, the application focus is in e-Health, specifically the retrieval of mental health records.**

***Keywords*-graph similarity; structure similarity; weight similarity; weighted Object Oriented RuleML; e-Health.**

I. Introduction

Semantic information can be represented using hierarchical structures, which express knowledge in multiple levels of detail. In the e-Business domain, vertex-labeled, edge-labeled and edge-weighted trees [1] are used in order to represent attributes of products. In [2], these weighted trees are generalized to weighted Directed Acyclic Graphs (wDAGs) in which substructures can be shared. Efficient similarity algorithms are required in many applications, such as for schema matching in databases, buyer-seller matching in e-Business, and health record retrieval in e-Health. They can also be used in social networks, e.g. to form similarity-clustered wellness or patient groups [3]. Calculating similarities between patient profiles (i.e., health records) is difficult, as the various aspects of a disease should be weighted differently, which entails that simple matching of attributes is not adequate in e-Health [4]. Weights are already used in similarity algorithms [1], [2], [4]. In [4], similar patients are identified based on similarity of symptoms and diseases. In this system, different aspects of a disease are weighted using regression estimation. Then, these calculated weights are used as coefficients in a weighted distance measure. Note that each particular user group (e.g., profiles of all patients having lung cancer) has the same values in the weight vector. This approach differs from the structural similarity algorithms [1], [2] which consider different set of weights for each profile (even if they belong to the same group). The similarity algorithms in [1], [2] compute the arithmetic mean of the two weights on corresponding edges of compared trees/wDAG in order to determine the weighted similarity. In this way, edge weights are used as scaling factors to ensure that the overall similarity value is in the real interval $[0, 1]$. We have found that this approach cannot differentiate trees nor wDAGs with different edge weights having identical or nearly identical structure similarity to the given query. Therefore, we propose modifications to the original weighted similarity algorithm to address this issue.

In this paper, a combined structure-weight similarity algorithm is proposed based on two component algorithms: a version of the structure similarity algorithm in [2] and a new weight similarity algorithm. In our approach, we perform ranked retrieval over a set of (meta)data represented as directed (vertex- and edge-)labeled (edge-)weighted graphs, each optionally associated with a data record. A special case is that the 'metadata' already are the 'data' to be retrieved, with no need for a separate data record. Similar to [2], graphs must be transformed to an internal representation before computing their similarity. Such graphs are expressed using a weighted extension of Object Oriented RuleML [5]. The XML parent-child structure reflects the hierarchical structure of the graphs, while the role element `<slot>` expresses edge labels and the attribute `weight` expresses edge weights. Also, the sharing of a rooted subgraph by multiple parents can be represented using a RuleML element with an XML `key` referred to from multiple `keyref`s. The graphs could be expressed using other representation approaches (e.g., Turtle [6] and RDF/XML [7]) as well. We assume that, given a query graph, a ranked list of matching (meta)data graphs (and consequently corresponding records), which are stored in a dataset, is constructed. The structure similarity and the weight similarity algorithms match the query graph

to each (meta)data graph and calculate their structure and edge weight similarity values, respectively. These pair values of structure and weight similarities (resulting from matching the query graph to each (meta)data graph) are considered as ranking criteria to generate the ranking list of (meta)data graphs. We demonstrate that this approach is able to differentiate the graphs having identical or nearly identical structure similarity but different edge-weight similarity to the given query.

The proposed combined structure-weight similarity approach is applied in e-Health domain. We represent (meta)data of Electronic Medical Records (EMRs) using graphs which express disorders and treatment priorities of patients. Then, our similarity approach is used to find mental health EMRs having similar (meta)data graphs to a given query. To provide patient privacy and security for health records as well as (meta)data, different technological safeguards as well as policies could be used [8]. In addition, using (meta)data could act as an extra level of privacy, as for extracting some statistics or trend, information in (meta)data itself is enough. Also, in retrieval applications, only records related to the ranked results would be retrieved not all records.

The rest of the paper is organized as follows. Section II explains our similarity approach. Section III focusses on an application of the proposed approach in the e-Health domain. Section IV concludes the paper.

II. Combined Structure-Weight Graph Similarity

In this section, graph representation and the architecture of the combined structure-weight similarity approach are presented. The theoretical basis of the proposed weight similarity is explained and the characteristics of the weight similarity are mentioned. A recursive weight similarity algorithm and the computational experiments on a synthetic dataset are presented.

A. Approach

Graph Representation: As stated earlier, we assume that we are given a set of records, with each record having an associated (meta)data represented as a graph. Note that all graphs throughout this paper are single-rooted wDAGs. All graphs are hierarchical as concepts can be represented using sub-concepts having different importance. The root vertex carries a class label, which types the main object. This object is further described by the labeled weighted edges leading to other labeled vertices of the graph, etc. Labels on outgoing edges from each given vertex are unique and appear in lexicographic (alphabetical) left-to-right order. Also, edge weights are values in the real interval $[0, 1]$ and for each graph its edge weights normalized; therefore, the sum of weights for all outgoing edges from each vertex equals 1. Further, we assume that given a query graph, a ranked list of the matching graphs is required to be constructed. Subsequently, these ranked (meta)data graphs are used to look up corresponding records. The computed weight similarity values should be comparable, therefore (similar to [1]) our graphs have to conform to the same standard schema.

Architecture: The proposed similarity approach has three modules: the structure similarity evaluation module, the weight similarity evaluation module, and the integration and ranking module (see Figure 1). We have a set of graphs $\mathbf{G} = \{G_1, G_2, G_3, \cdots, G_n\}$, which represents the (meta)data for a set of records. Both number of vertices and edges are assumed to be finite. Given a graph G', the structure similarity of G' with each member of $\mathbf{G}$ is calculated using the recursive graph similarity algorithm proposed in [2]; here G' may represent a query. The structure similarity algorithm is iterative. The given graphs are traversed from their roots to their leaves (top-down) and then their similarity is computed bottom-up. The structure similarity values and weight similarity values are in the real interval $[0, 1]$. The weight similarity evaluation module matches each member of $\mathbf{G}$ with G'; then it calculates the edge-weight similarity value. Figure 1 shows the architecture of the similarity approach where $\mathbf{G}$ and G' represent a set of graphs and a given query graph being matched, $sSim(\mathbf{G}, G')$ denotes their structure similarity values, and $wSim(\mathbf{G}, G')$ expresses their weight similarity values.

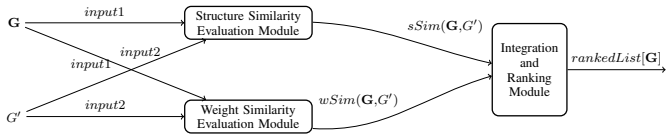

Figure 1: Proposed Combined Structure-Weight Similarity Architecture

The structure similarity values and weight similarity values of $\mathbf{G}$ and G' are inputs to the integration and ranking module. After receiving the similarity pairs $[sSim(G_i, G'), wSim(G_i, G')]$ for all graphs $i = \{1, 2, 3, \cdots, n\}$ in set $\mathbf{G}$, the integration and ranking module ranks the graphs in $\mathbf{G}$ based on the structure similarity and weight similarity. Structure and weight similarity values could be combined with different approaches. Here, we consider weight similarity as the secondary criterion in ranking of graphs. As a result, G_1 could appear before G_2 ($G_1 \succ G_2$) in the ranked list if and only if structure similarity value of G_1 to G' (the query) is greater than the structure similarity value of G_2 to G'; or the difference between their structure similarity is less than or equal to a threshold while the weight similarity value G_1 to G' is greater than the weight similarity value of G_2 to G'. Thus,

(a) $G_1 \succ G_2$ if and only if $[sSim(G_1, G') > sSim(G_2, G')]$, or $[|sSim(G_1, G') - sSim(G_2, G')| \leq$

$Threshold$ and $wSim(G_1, G') > wSim(G_2, G')$].

(b) $G_1 \succ G_2$ or $G_2 \succ G_1$ if $[|sSim(G_1, G') - sSim(G_2, G')| \leq Threshold$ and $wSim(G_1, G') = wSim(G_2, G')]$

In this paper, we consider the threshold equal to 0. For each graph, we keep a count of the number of edges, assigning a unique integer *j* to each edge, starting from 1 in top-down (root to leaf) and left-to-right order. As a result, each edge is represented by e_j, $j \in \{1, 2, 3, \cdots, z\}$, considering z as the total number of edges in a graph. As all edges are directed, the source vertex u and the destination vertex v of each edge e_j can be represented as an ordered pair (u, v). Also, the weight of edge e_j is represented as $w(e_j)$. The edge e_j in graph G and the edge $e_{j'}$ in graph G' are called *corresponding edges* if and only if they have identical edge labels as well as identical source vertex labels and destination vertex labels. The relation between corresponding edges e_j and $e_{j'}$ is denoted as $e_j \doteq e_{j'}$. Consider d_u as the depth of vertex u. In our graphs, d_u and $d_{u'}$ are equal for two corresponding edges e_j and $e_{j'}$.

B. Weight Similarity

In the proposed weight similarity approach, the similarity of weights related to two corresponding edges can be calculated based on two similarity measures [9], viz. Manhattan distance, Equation 1, or Min/Max similarity measure, Equation 2, as given below:

$$weSim1 = 1 - |w(e_j) - w(e_{j'})| \tag{1}$$

$$weSim2 = \frac{\min(w(e_j), w(e_{j'}))}{\max(w(e_j), w(e_{j'}))} \tag{2}$$

The importance of each edge can be considered to be a function of the depth of its source vertex. As stated earlier, the root vertex carries a class label, which types the main object; therefore, the outgoing edges from the root have the highest importance. This importance decreases as the depth of the source vertex of the edge increases. Similarly, contribution of the weight similarity of two corresponding edges in weight similarity of two graphs depends on the depth of the source vertex related to corresponding edges. The coefficient for adjusting the contribution of edge weight similarity needs to decreases as the depth of the source vertex of corresponding edges increases. One approach for defining this coefficient is using an exponential function with D as the fixed base and $d + 1$ as the variable exponent. Therefore, in this paper, the adjustment coefficient is expressed as D^{d+1}. If p enumerates the pairs of corresponding edges in depth d and m_d ($m_d \geq 0$) denotes the number of corresponding edges in depth d, the weight similarity value of graphs is expressed using Equation 3. In this equation, each edge similarity value is multiplied by D^{d+1}, in which D is the global depth degradation factor ($D \leq 0.5$) and d is the depth of the source vertex of the edge. $0 \leq d \leq d_{max}$, where d_{max} is the maximum possible depth of the source vertex of corresponding edges in two graphs.

$$Sim = \sum_{d=1}^{d_{max}} (\sum_{p=1}^{m_d} weSim_p \cdot D^{d+1}) \tag{3}$$

As the similarity of weights, numbers in the real interval $[0, 1]$, related to two corresponding edges is calculated using the Manhattan distance (Equation 1) or the Min/Max similarity measure (Equation 2), the similarity value of a pair of weights $weSim_p$, $p \in \{1, 2, 3, \cdots, m_d\}$ is in interval $[0, 1]$. Also d, which is the depth of the source vertex related to an edge, could be a value larger than or equal to 0. As a result, D^{d+1} is a positive number. Thus, the summation of $(weSim_p \cdot D^{d+1})$ for all corresponding edges could result in a value larger than 1 and therefore Sim could be greater than 1. In order to express the graph similarity as a value in real interval $[0, 1]$, the combined edge weight similarity values (viz. Sim) is normalized by the sum of the D^{d+1} used in various iterations of the recursive weight similarity algorithm. Starting from the first level in graphs, each time a pair of weights is compared, the related depth factor is added and this process is repeated for all levels of graphs. The normalization factor denoted by F is expressed as,

$$F = \sum_{d=1}^{d_{max}} (\sum_{p=1}^{m_d} D^{d+1}) \tag{4}$$

Thus, the normalized weight similarity of two graphs ($wSim$) is given as,

$$wSim = \frac{Sim}{F}, \tag{5}$$

which lies in real interval $[0, 1]$. The global depth degradation factor (D) could be equal to 1. In this case, the proposed similarity approach gives the same importance to the weight similarities of various levels of the graphs and the arithmetic mean of the weight similarity values is calculated. Therefore, the result of such a calculation is identical to considering the weight similarity of all attributes having the same effect on the weight similarity of two graphs. This approach results in a linear trend of similarity values. In Equation 6, m_{total} denotes the number of corresponding edges in total. $weSim1(w(e_j), w(e_{j'}))$ is the similarity of weights related to two corresponding edges based on the Manhattan distance, while $wSim$ is the global weight similarity of two graphs based on the Manhattan distance. The same relation holds when the weight similarity is calculated based on the Min/Max similarity measure as well.

$$wSim = (1/m_{total}) \cdot \sum_{k=1}^{m_{total}} (weSim1(w(e_j), w(e_{j'}))) \tag{6}$$

The weight similarity also has the following characteristics: (a) The similarity value generated by the weight similarity

approach is a non-negative number. The minimum similarity value equals 0. (b) The weight similarity of a graph to itself is 1.0. The similarity of each pair of weights $weSim(w(e_j), w(e_{j'}))$ is 1.0. Therefore, Sim has the same value as F and as a result the weight similarity of two graphs (i.e., $wSim$) is equal to 1.0. (c) The weight similarity measure is a symmetric function, as the order of pair of graphs does not affect the result of the computation of weight similarity. (d) The weight similarity like many other similarities does not obey triangular inequality. The weight similarity measure is a partial matching approach as only the weights related to the corresponding edges are compared.

C. Algorithm

Algorithm 1, which calculates the weight similarity of two graphs based on Manhattan distance, is represented in Figure 2.

```
1:  procedure WSIMILARITY(G, G')
2:      if G or G' only contains a single vertex then
3:          return 0
4:      end if
5:      if G.root.label ≠ G'.root.label then
6:          return 0
7:      else
8:          d ← root(G).depth
9:          k ← 1
10:         k' ← 1
11:         while k ≤ G.root.outDegree
                        ∧ k' ≤ G'.root.outDegree do
12:             e_j ← G[k].root.edge
13:             e_j' ← G'[k'].root.edge
14:             if e_j ≐ e_j' then
15:                 F ← F + D^(d+1)
16:                 weSim ← (1 − |w(e_j) − w(e_j')|)
17:                 Sim ← Sim + weSim · D^(d+1)
                        + wSimilarity(G.subgraph(e_j),
                                      G'.subgraph(e_j'))
18:                 k ← k + 1
19:                 k' ← k' + 1
20:             else if e_j ≻ e_j' then
21:                 k ← k + 1
22:             else
23:                 k' ← k' + 1
24:             end if
25:         end while
26:         wSim ← Sim/F
27:         return wSim
28:     end if
29: end procedure
```

Figure 2: Algorithm 1. Weight Similarity of two Graphs based on Manhattan Distance

Algorithm 1 (see Figure 2) gives the weight similarity algorithm, which traverses two input graphs G and G' in a left-right depth-first strategy. The parameter of the algorithm is D, which represents the global depth degradation factor. Here we assume that D is equal to 0.5; however, a learning component could be used to adjust the parameter. Considering graphs G and G' as the inputs of the algorithm, $G.subgraph(e_j)$ denotes the sub-graph rooted at destination vertex of e_j in graph G. $G.root.label$, $G.root.inDegree$, and $G.root.outDegree$ represent vertex label, in-degree, and out-degree of the root of graph G, respectively. Also, $e_j \succ e_{j'}$ represents that e_j could appear before $e_{j'}$ in a lexicographic ordered list. $weSim$ is the similarity of weights related to two corresponding edges. $root(G).depth$ is a function which gives the depth for root of graph G relative to the root of the original graph. The output, $wSim$, is the weight similarity value of G and G'.

The proposed weight similarity algorithm traverses two given graphs 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 is calculated using Equation 1 or 2. Two pointer variables, k and k', indicate the positions of two outgoing edges being matched. If $e_j \succ e_{j'}$, k is set to point to the next outgoing edge in G, while if $e_{j'} \succ e_j$, k' would be increased to point to the next outgoing edge in G'. If $e_j \doteq e_{j'}$, k and k' are set to point to the next outgoing edges in G and G', respectively. The loop is terminated as soon as any one of the following conditions is met: $k > G.root.outDegree$ or $k' > G'.root.outDegree$.

The algorithm is recursive, so the base case and recursive case should be defined. The base case is where the problem can be solved directly, while in the recursive case the problem is expressed as subproblems that are closer to the base case [10, pp. 228]. In this algorithm the base of the recursion is where G or G' only contains a single vertex (Algorithm 1, lines 2-4) or if $G.root.label \neq G'.root.label$ (Algorithm 1, lines 5-6). In both cases, their weight similarity is 0. The algorithm is tail recursive, i.e., the recursive invocation is the very last thing which is performed [10, pp. 245]. In the recursive case, the algorithm recursively invokes itself using the roots of two sub-graphs of G and G' as arguments (Algorithm 1, line 17).

As stated earlier, the labels of outgoing edges from each vertex are arranged in the lexicographic order. Also, two pointers indicate the positions of two edges being matched. Using these features, the time complexity of the algorithm is improved. If G or G' only contains a single vertex or $G.root.label \neq G'.root.label$ for the roots of two graphs, then the algorithm sets the weight similarity directly to 0 without any further computation; If $G.root.label = G'.root.label$, the algorithm uses one loop (Algorithm 1, line 11) to find the corresponding edges. For two graphs, consider t, $t \in \{1, 2, 3, \cdots, r\}$, in which r equals to the total number of pairs of matched non-leaf vertices. When matching all outgoing edges of a pair of vertices, three cases should be considered: (i) If $e_j \succ e_{j'}$ or $e_j \doteq e_{j'}$, for all values of k and k', the number of iterations equals to ${I_G}^t = G.root.outDegree$, (ii) If

$e_{j'} \succ e_j$, for all values of k and k', the number of iterations is equal to ${I_{G'}}^t = G'.root.outDegree$, and (iii) If only for some values of k and k', $e_j \succ e_{j'}$ or $e_j \doteq e_{j'}$, the number of iterations to find the corresponding edges is in the interval $[\min({I_G}^t, {I_{G'}}^t), \max({I_G}^t, {I_{G'}}^t)]$. The number of iterations for finding all corresponding edges in graphs, I, equals to the summation of iterations performed for each pair of vertices; I is in interval $[\sum_{t=1}^{r} \min({I_G}^t, {I_{G'}}^t), \sum_{t=1}^{r} \max({I_G}^t, {I_{G'}}^t)]$. In the worst case, $I = \sum_{t=1}^{r} \max({I_G}^t, {I_{G'}}^t)]$, and therefore the complexity of the algorithm is $\Theta(\sum_{t=1}^{r} \max({I_G}^t, {I_{G'}}^t))$.

D. Computational Experiments

Now, we test the proposed weight similarity algorithm on a synthetic dataset, in which weights are changed systematically to understand the effects of structure and weights on the similarity. The dataset contains graphs structurally identical to the graphs given in Figure 3, but with different weights. The graphs are balanced with maximum breadth assuming branching factor of 2. The dataset contains 29 graphs.

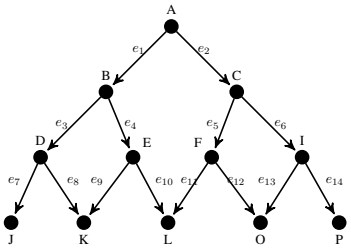

Figure 3: Graph Structure of Metadata in Dataset

In this dataset, we have five possible values for a pair of edge weights: $[0.01, 0.99]$, $[0.25, 0.25]$, $[0.5, 0.5]$, $[0.75, 0.25]$, or $[0.99, 0.01]$. In G_1 of dataset, weights of all edges having the same source vertex are $[0.01, 0.99]$. Now, we change the edge weights from right to left in a level and then bottom-up for various levels, exhausting the five possible sets of edge weight pairs. This results in 29 graphs in the dataset, of which eight graphs, G_1 to G_8, are shown in Table I, where each row represents the weights related to a graph [1]. Enabling a compact specification and description of the weights, this notation is used to illustrate different weight values for one graph structure.

Considering this systematic changes in weights, the weight similarities of G_1 in the dataset with respect to the remaining graphs are expected to decrease gradually. Therefore, the synthetic dataset provides a starting point for an evaluation of our weight similarity algorithm.

[1]The complete dataset is available from authors.

Table I: Edge Weights of a Subset (G_1 to G_8) of 29 Graphs (G_1 to G_{29}) with the Structure given in Figure 3

((a)) Weights of Edges e_1 to e_7

Graph	$w(e_1)$	$w(e_2)$	$w(e_3)$	$w(e_4)$	$w(e_5)$	$w(e_6)$	$w(e_7)$
G_1	0.01	0.99	0.01	0.99	0.01	0.99	0.01
G_2	0.01	0.99	0.01	0.99	0.01	0.99	0.01
G_3	0.01	0.99	0.01	0.99	0.01	0.99	0.01
G_4	0.01	0.99	0.01	0.99	0.01	0.99	0.01
G_5	0.01	0.99	0.01	0.99	0.01	0.99	0.25
G_6	0.01	0.99	0.01	0.99	0.25	0.75	0.25
G_7	0.01	0.99	0.25	0.75	0.25	0.75	0.25
G_8	0.25	0.75	0.25	0.75	0.25	0.75	0.25

((b)) Weights of Edges e_8 to e_{14}

Graph	$w(e_8)$	$w(e_9)$	$w(e_{10})$	$w(e_{11})$	$w(e_{12})$	$w(e_{13})$	$w(e_{14})$
G_1	0.99	0.01	0.99	0.01	0.99	0.01	0.99
G_2	0.99	0.01	0.99	0.01	0.99	0.25	0.75
G_3	0.99	0.01	0.99	0.25	0.75	0.25	0.75
G_4	0.99	0.25	0.75	0.25	0.75	0.25	0.75
G_5	0.75	0.25	0.75	0.25	0.75	0.25	0.75
G_6	0.01	0.25	0.75	0.25	0.75	0.25	0.75
G_7	0.01	0.25	0.75	0.25	0.75	0.25	0.75
G_8	0.25	0.75	0.25	0.75	0.25	0.75	0.25

Figure 4 depicts the similarity values of G_1 for the synthetic dataset with the remaining graphs using the graph similarity algorithm in [2] as well as our combined structure-weight similarity algorithm (using the similarity measure based on the Manhattan distance).

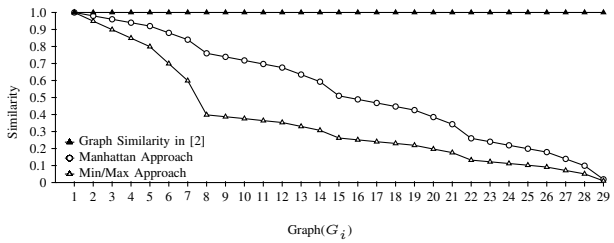

Figure 4: Similarity of G_1 to 29 Graphs in the Dataset

While similarity values based on the previous graph similarity algorithm [2] are always equal to 1, our combined structure-weight similarity approach differentiates the structurally identical graphs with different weights. Also, Figure 4 gives a comparison of the two similarity measures, the similarity measure based on the Manhattan distance and the Min/Max similarity measure. Here again we compute the similarity w.r.t. G_1. For the depth degradation factor equal to 0.5 and for the same set of weights for a dataset, both similarity measures generate similarity values with a decreasing trend. It is important to note that for Figure 4, the similarity decreases as a result of the systematic change of weights of edges (having the same source): gradual increase of the edge weight for the left vertex and gradual decrease of the edge weight for the right vertex. The bumps in the similarity plots (e.g. at G_8, G_{15}, and G_{22}) are observed as the result of level transitions, i.e., the systematic changes of weights in each level of the graph.

We have given above the computational results for the similarities of members of a dataset. We can generalize the behavior of the similarity computation to other possible graph structures such as trees [1], and generalized trees [11], and conclude that the proposed weight-similarity algorithm, with any one of the similarity measures, is effective in differentiating graphs having identical or nearly identical structure similarity values (but different weights). Weight similarity considers only weight of common subgraphs of two graphs being compared, while structure similarity takes into account common as well as uncommon subgraphs. Therefore, two graphs could be similar from weight similarity perspective, while their uncommon sub-graphs are large (i.e. small structure similarity). Note that although the numerical similarity values of the two similarity measures are different, they result in the same relative ranking of the graphs for the given query. Since there is no universal benchmark for evaluating similarity [12], it is not possible to select or recommend one of the similarity measures over the other and both similarity measures could be used for the purposes of relative ranking.

III. E-Health Application: Mental Health Electronic Medical Record

Group therapy is used as a treatment option for drug abusers [13, pp. 577-620]. Newcomers should be placed in groups with at least one or two similar members. Open group membership in which new members are allowed to enter as others leave is the norm [14, pp. 262-273]. Therefore, retrieving similar mental health EMRs to select patients for group therapy is a challenging task. This selection should be based on the gathered dynamic, behavioral, and diagnostic information in a screening interview [15, pp. 934]. Consider the scenario where the user (e.g., a psychologist) wants to find an appropriate group for a new patient in order to schedule group therapy sessions. In this case, mental health EMRs that describe similar disorders as well as treatment priorities should be found. Each (meta)data expresses the individualized treatment plan about patient's disorders and the treatment priorities based on the last psychological evaluation. Similar to [2], we represent the attributes of each (meta)data using a graph based on a standard schema. The attributes of this schema are extracted from [15], [16], and the terms representing the (meta)data are based on DSM IV [16]. The attributes express possible affective, behavioral, and cognitive problems of a patient. The edge weights in graphs represent the relative priority regarding treatment of each disorder in the group therapy session. Therefore, severe, influential, and dangerous disorders as well as the items for which treatments have the greatest benefit have higher priority (i.e., higher weight) in our treatment-oriented (meta)data. As treatment priorities change over time, edge weights could be different in each evaluation phase by the psychologist. In order to select patients for group therapy, in the proposed system the edge weights of (meta)data are always related to the last psychological evaluation of patients (available in the mental health records). Figure 5 illustrates the generic structure of (meta)data of mental health EMRs in the database as well as a query having the same structure.

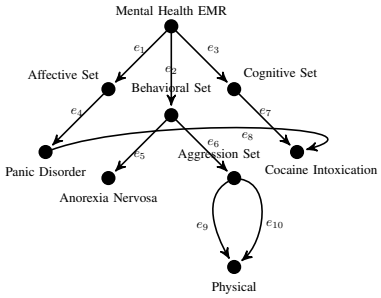

Figure 5: Graph Structure of a Query and Metadata of Mental Health EMRs

Table II represents the edge labels of the generic structure (in Figure 5), in which $l(e_j)$ denotes the label of edge e_j, $j \in \{1, 2, 3, \cdots, 10\}$. The patients have panic disorder and also delirium due to cocaine intoxication. Other disorders of the patients are anorexia nervosa and physical aggression including fantasies and real acts [15, pp. 421].

Table II: Edge Labels of a Query and Metadata Graphs (having the Structure in Figure 5) for Mental Health EMRs

$l(e_1)$ Affective disorders	$l(e_6)$ Aggression
$l(e_2)$ Behavioral disorders	$l(e_7)$ Delirium
$l(e_3)$ Cognitive disorders	$l(e_8)$ Substance induced panic
$l(e_4)$ Anxiety	$l(e_9)$ Real act
$l(e_5)$ Appetite disorder	$l(e_{10})$ Fantasies

Edge weights of four EMR (meta)data, representing the diagnosis segment of a mental health EMR, and a query are illustrated in Table III. Note the different last subscripts for the two edges emanating from the Aggression vertex and terminating at the same Physical destination vertex. Further, there are three edges from the root vertex.

Table III: Edge Weights of a Query (G'_1) and four Metadata Graphs (having the Structure in Figure 5) for Mental Health EMRs

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

Now we compare the similarity of query with the four (meta)data graphs G_1, G_2, G_3, and G_4 of the EMRs given in Tables II and III using the combined structure-weight similarity algorithm. The computed similarity values are given in Table IV. The structure similarity values between

query G' and any of four (meta)data graphs are identical; therefore, we cannot distinguish between them using the structure similarity alone. The edge weight similarity results using the proposed algorithm are also shown in Table IV.

Table IV: Computational Results for the Metadata of Mental Health EMRs and the Query in Table III

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

We can clearly see that the similarities are different and they can be used to rank four (meta)data graphs. Further, both similarity measures (see columns 4 and 5 in Table IV) are equally acceptable as they result in the same relative ranks. Instead of ranked graphs based on their similarity to a given query, the proposed approach could cluster the mental health EMRs based on a threshold to facilitate creation of supportive virtual communities, which is one of the main goals of Health 3.0 [17].

IV. Conclusion

Our combined structure-weight similarity approach is able to distinguish graphs having identical or nearly identical structure but different weights. By considering the weight similarity in addition to the structure similarity, preferences of user are compared with the preferences expressed as edge weights of graphs stored in dataset. The similarity of edge weights is calculated in a recursive way, giving more importance to weights of edges in higher levels of a graph. The combined structure-weight similarity algorithm has been implemented in Java and it has been applied to retrieve mental health electronic medical records (EMRs).

Acknowledgments

The authors would like to thank Mehrdad Kiani, M.D., who provided help in the health domain. This research is partially funded by a Discovery Grant from the Natural Sciences and Engineering Council of Canada and a Ph.D. Fellowship Grant from the Atlantic Computational Excellence Network (ACEnet) awarded to the second author.

References

[1] Bhavsar, V.C. and Boley, H. and Yang, L., "A Weighted-Tree Similarity Algorithm for Multi-Agent Systems in E-Business Environments," *Computational Intelligence*, vol. 20, no. 4, pp. 584–602, 2004.

[2] Jin, J., "Similarity of Weighted Directed Acyclic Graphs," MSc Thesis, Faculty of Computer Science, University of New Brunswick, Canada, Sep. 2006.

[3] Boley, H. and Shafiq, O., and Smith, D. and Osmun, T., "The Social Semantic Subweb of Virtual Patient Support Groups," in *Proc. the 3rd Canadian Semantic Web Symposium (CSWS2011), Vancouver, British Columbia, Canada*. CEUR, Aug. 2011, pp. 1–18.

[4] Fritz, P. and Klenk, S. and Dippon, J. and Heidemann, G., "Determining patient similarity in medical social networks," in *Proc. MedEx Workshop*, 2010, pp. 6–13.

[5] H. Boley, "Object-Oriented RuleML: User-Level Roles, URI-Grounded Clauses, and Order-Sorted Terms," in *Proc. Rules and Rule Markup Languages for the Semantic Web (RuleML-2003)*. LNCS 2876, Springer, Oct. 2003, pp. 1–16.

[6] D. Beckett and T. Berners-Lee. (2011) Turtle A Readable RDF Syntax. [Online]. Available: http://www.w3.org/TeamSubmission/turtle/

[7] Cyganiak, R. and Wood, D., Ed., *Resource Description Framework (RDF): Concepts and Abstract Syntax*. World Wide Web Consortium, Jan. 2013. [Online]. Available: http://www.w3.org/TR/2013/WD-rdf11-concepts-20130115/

[8] Jacques, L.B., "Electronic Health Records and Respect for Patient Privacy: A Prescription for Compatibility," *Vand. J. Ent. & Tech. L.*, vol. 13, pp. 441–462, 2011.

[9] Boriah, S. and Chandola, V. and Kumar, V., "Similarity Measures for Categorical Data: A Comparative Evaluation," in *Proc. the 8th SIAM International Conference on Data Mining*, 2008, pp. 243–254.

[10] Drake, P., *Data Structures and Algorithms in Java*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2005.

[11] Dehmer, M. and Emmert-Streib, F. and Kilian, J., "A Similarity Measure for Graphs with Low Computational Complexity," *Applied Mathematics and Computation*, vol. 182, no. 1, pp. 447 – 459, 2006.

[12] Janowicz, K. and Raubal, M. and Schwering, A. and Kuhn, W., "Semantic Similarity Measurement and Geospatial Applications," *T. GIS*, vol. 12, no. 6, pp. 651–659, 2008.

[13] Carr, A., *The Handbook of Child and Adolescent Clinical Psychology: A Contextual Approach*. Rouledge, New York: Taylor and Francis Group, 1999.

[14] Ruiz, P. and Strain, E.C. and Langrod, J., *The Substance Abuse Handbook*, ser. Doody's all reviewed collection. Philadelphia: Wolters Kluwer Health/Lippincott Williams and Wilkins, 2007.

[15] Sadock, B.J. and Sadock, V.A., *Kaplan and Sadock's Synopsis of Psychiatry: Behavioral Sciences/Clinical Psychiatry*, 10th ed. Philadelphia: Lippincott Williams and Wilkins, 2007.

[16] A. P. Association and A. P. A. T. F. on DSM-IV., *Diagnostic and Statistical Manual of Mental Disorders: DSM-IV-TR.*, 4th ed., ser. Diagnostic and Statistical Manual of Mental Disorders. Washington, DC: American Psychiatric Association, 2000.

[17] Kiani, M. and Bhavsar, V.C. and Boley, H., "Clustering Using Combined Structure-Weight Graph Similarity," University of New Brunswick, Canada, Internal Report (In Preparation).