Classification of large Graphs by a local Tree decomposition

(DMIN'05)

Frank Emmert-Streib*, Matthias Dehmer†, Jürgen Kilian†‡,
*Stowers Institute for Medical Research
1000 E. 50th Street, Kansas City, MO 64110, USA
Email: fes@stowers-institute.org
†Technical University Darmstadt
64289 Darmstadt, Germany
Email: dehmer@informatik.tu-darmstadt.de
‡Email: kilian@noteserver.org

Abstract— **We present a binary graph classifier (BGC) which allows to classify large, unweighted, undirected graphs. This classifier is based on a local decomposition of the graph for each node in generalized trees. The obtained trees, forming the tree set of the graph, are then pairwise compared by a generalized tree-similarity-algorithm (GTSA) and the resulting similarity scores determine a characteristic similarity distribution of the graph. Classification in this context is defined as mutual consistency for all pure and mixed tree sets and their resulting similarity distributions in a graph class. We demonstrate the application of this method to an artificially generated data set and for data from microarray experiments of cervical cancer.**

I. Introduction

The problem of graph similarity and the structural comparison of graphs is an interesting and important question in many areas of science, e.g. biology [11] and chemistry [17]. Measures of distances between graphs [3] have been frequently investigated. KADEN [10] distinguished the following principles for the definition of distances between graphs (by):

- a minimal number of changes
- maximal matches
- graph grammars
- which definitions are based on different principles

Thereby many distances on graphs are based on isomorphic relations and subgraph isomorphism [14], [20], respectively. An example of such a graph distance is the well-known ZELINKA-distance [23]. The ZELINKA-distance is based on the principle that two graphs are more similar, the bigger the common induced subgraph is. In other words, graphs which have a large common induced subgraph have a small distance and vice versa. ZELINKA was the first to introduce this measure for unlabeled graphs. SOBIK [18], [19] and KADEN [8], [9] generalised this measure for arbitrary (also labeled) graphs of different order and proved that it is a metric. KADEN has obtained further similarity measures on graphs by transforming the graphs by injective mapping. For example, KADEN [8] considered *line graphs* [1] and used the ZELINKA-distance to compute the similarity of the transformed graphs. Because the subgraph isomorphism problem is NP-complete [20], the complexity of those measures is considered to be unacceptable for practical use. Figure (I) shows the number of isomorphism classes for undirected Graphs [7]. SHAPIRO [16] introduced a well-known similarity metric for graphs based on the corresponding adjacency matrices. Let the graphs be G_1, G_2 and the corresponding adjacency matrices A_1, A_2. Permute the rows and the columns in the matrix A_2 in such a way that the matrix elements coincide with the matrix elements of A_1 as much as possible. Then, SHAPIRO defines the graph distance between G_1, G_2 by the minimal number of dissenting matrix elements and proves that it is a metric.

In addition, an important class of similarity measures based on the edit distance of graphs has been investigated by [3], [24], [25]. The edit distance is based on basic weighted transformation steps, like deletions, substitutions, and insertions of vertices and edges. Since there is an infinite number of different possibilities for transforming G_2 in G_1, the similarity of the graphs is defined as the minimum cost of transformations. Other approaches to inexact graph matching consider distances between graphs on the basis of *graph grammars* [6]. These methods are primarily interesting for theoretical aspects but not for practical use, since the specific grammar is difficult to obtain.

In this work we first introduce a similarity measure for a special class of graphs: *unlabeled, hierarchic, and directed graphs* which we call in the following *generalized trees*. The main idea of this similarity measure is based on the derivation of property strings for each generalized tree and then to align the property stings representing the trees by a *dynamic programming* [2] technique. From the resulting alignment one obtains a value of the scoring function which is minimized during the alignment process. The similarity of two generalized trees will be expressed by a cumulation of local similarity functions which weighs two types of alignments: *outdegree* and *indegree* alignments on a generalized tree level. In section (II) we will give a mathematical motivation of the method. In section (III) we will explain the construction

TABLE I

SET OF ISOMORPHISM CLASSES FOR UNDIRECTED GRAPHS

$\|V\|$	Number of pairwise non isomorphic Graphs
1	1
2	2
3	4
4	11
5	34
6	156
7	1 044
8	12 346
9	271 346
10	12 005 108
11	1 018 997 864
12	165 091 172 592

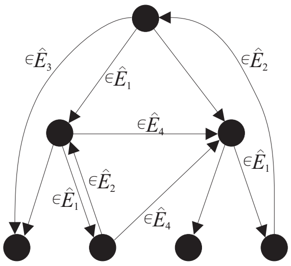

Fig. 1. Edge types of generalized trees.

of the similarity measure in detail. We call this method the generalized tree-similarity-algorithm (GTSA). In section (IV) we will use the GTSA to define a binary graph classifier (BGC) for undirected, unweighted graphs by decomposing a graph locally in trees. In the results section (V) we present first results by applying the BGC to two different data sets consisting of graphs of the order 10^3 and 10^4.

II. STRUCTURAL ASPECTS OF GENERALIZED TREES

In the introduction (I), we mentioned some known methods in order to determine the similarity between graphs. But most methods are not suitable for practical use, because the complexity is considered to be unacceptable for measuring graphs of large order. A similarity measure between generalized trees should have the following properties:

- The similarity measure should be efficient in terms of mass data
- The formalization of the similarity measure should be plausible
- Losing of structure while measuring the similarity between two generalized trees should be as minimal as possible
- A weighting of specific structural properties (e.g., specific edge relations) is desirable

In order to motivate our method we will now state the class of generalized trees which has been introduced by MEHLER et al. [13]. Furthermore we examine some properties of degree sequences of these trees.

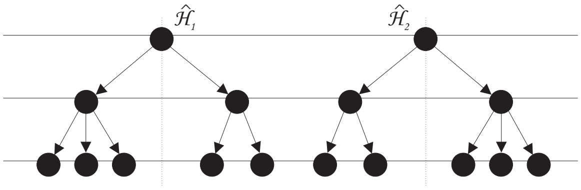

Fig. 2. Two asymmetric trees $\hat{\mathcal{H}}_1$ and $\hat{\mathcal{H}}_2$ with the same degree sequences.

Definition 2.1: Let be the vertex set

$$\hat{V} := \{v_{0,1}, v_{1,1}, v_{1,2}, \dots, v_{1,\sigma_1}, v_{2,1},$$
$$v_{2,2}, \dots, v_{2,\sigma_2}, \dots, v_{h,1}, v_{h,2}, \dots, v_{h,\sigma_h}\}$$

Define h as the maximal length of a path from the root $v_{0,1}$ to a leaf. $v_{i,j}$ denotes the j-th vertex on the i-th level, $0 \leq i \leq h$, $1 \leq j \leq \sigma_i$. σ_i is maximal in the sense that there is no other vertex sequence such that $v_{i,1}, v_{i,2}, \dots, v_{i,\hat{\sigma}_i}$ with $\hat{\sigma}_i > \sigma_i$. The edge set (see Figure (1)) $\hat{E}$ is defined by [13]

- $(\hat{E}_1)$ *Kernel edges*: the *kernel hierarchy* is induced by the *Kernel edges*. *Kernel edges* associate dominating nodes with their immediately dominated successor nodes.
- $(\hat{E}_2)$ *Up edges* associate analogously nodes of the kernel hierarchy with one of their (dominating) predecessor nodes.
- $(\hat{E}_3)$ *Down edges* associate nodes of the kernel hierarchy with one of their (dominated) successor nodes in terms of that kernel hierarchy.
- $(\hat{E}_4)$ *Cross edges* associate nodes of the kernel hierarchy, none of which is an (immediate) predecessor of the other in terms of the kernel hierarchy.

Then $\hat{\mathcal{H}} := (\hat{V}, \hat{E})$, $\hat{E} := \hat{E}_1 \cup \hat{E}_2 \cup \hat{E}_3 \cup \hat{E}_4$ denotes the unlabeled, hierarchic, and directed graph which we call a generalized tree or just tree when no conflict occurs.

Definition 2.2: Let $\hat{\mathcal{H}} = (\hat{V}, \hat{E}), |V| < \infty$ be a generalized tree.

$$\mathcal{N}^+(v) := \{\tilde{v} \in \hat{V} \backslash \{v\} | (v, \tilde{v}) \in \hat{E}\}$$
$$\mathcal{N}^-(v) := \{\tilde{u} \in \hat{V} \backslash \{v\} | (\tilde{u}, v) \in \hat{E}\}$$
$$\delta_{out}(v) := |\mathcal{N}^+(v)|,$$
$$\delta_{in}(v) := |\mathcal{N}^-(v)|.$$

$\mathcal{N}^+(v)$ and $\mathcal{N}^-(v)$ denotes the set of out-neighbors of v and the set of in-neighbors of v, respectively. $s_j^{out}(\hat{\mathcal{H}}) \in \mathbb{N}, 0 \leq j \leq k_{out} := \max_{v \in V}\{\delta_{out}(v)\}$ (or $s_i^{in}(\hat{\mathcal{H}}) \in \mathbb{N}, 0 \leq i \leq k_{in} := \max_{v \in V}\{\delta_{in}(v)\}$) denotes the number of vertices of $\hat{\mathcal{H}}$ with outdegree j (or indegree i). The vector $s^{out}(\hat{\mathcal{H}}) := (s_0^{out}(\hat{\mathcal{H}}), s_1^{out}(\hat{\mathcal{H}}), \dots, s_{k_{out}}^{out}(\hat{\mathcal{H}}))$ or $s^{in}(\hat{\mathcal{H}}) := (s_0^{in}(\hat{\mathcal{H}}), s_1^{in}(\hat{\mathcal{H}}), \dots, s_{k_{in}}^{in}(\hat{\mathcal{H}}))$ is called the outdegree (or indegree) sequence of $\hat{\mathcal{H}}$.

Now if we simply compare the outdegree and indegree sequence of two trees which fulfill Definition (2.1), it is evident that the concept of degree sequences is only restricted applicable for matching our trees, because the outdegree and indegree sequences includes only the number of vertices with out/(in)degree $j/(i)$. We provide now an example of two generalized trees which are shown in Figure (2). Clearly, their topology can not adequately described by the outdegree and indegree vectors. By definition (2.2), the trees in figure (2) have the same outdegree and indegree sequences, $s^{out}(\hat{\mathcal{H}}_1) = (5,0,2,1) = s^{out}(\hat{\mathcal{H}}_2) = (5,0,2,1) \wedge s^{in}(\hat{\mathcal{H}}_1) = (1,7) = s^{in}(\hat{\mathcal{H}}_2) = (1,7)$. But they are not symmetrically located on the symmetry axis, indicated by the dashed lines. Therefore, we see that degree sequences of the given graphs have no influence on the embeddings of substructures in the tree.

If we consider degree sequences in terms of isomorphic relations, we express the simple Proposition.

Proposition 2.3: Let $\hat{\mathcal{H}}_1$ and $\hat{\mathcal{H}}_2$ be generalized trees. If ϕ is a graph isomorphism of $\hat{\mathcal{H}}_1$ on $\hat{\mathcal{H}}_2$, it holds for $1 \leq i \leq |\hat{V}|$

$$\delta_{out}(v_i) = \phi(\delta_{out}(v_i))$$

and

$$\delta_{in}(v_i) = \phi(\delta_{in}(v_i)).$$

But from Proposition (2.3) we conclude immediately, that trees which are isomorphic have the same outdegree and indegree sequences. However, the reverse assertion is not always true.

III. Constructing similarity measures for generalized tree matching

In section (II), we saw that degree sequences cannot describe the topology of a graph completely. Since we are examining generalized trees, we will look, in the following, at the outdegree and indegree sequences (on a level i), induced by the vertex sequences $v_{i,1}, v_{i,2}, \ldots, v_{i,\sigma_i}$ and their edge relations in terms of definition (2.1), see Figure (3).

Now, the more similar with respect to a *cost function* α the outdegree and indegree sequences on the levels $i, 0 \leq i \leq h$ are, the more similar is the common structure of the trees. Define $w_k^{\hat{\mathcal{H}}^k} := v_{0,1}^{\hat{\mathcal{H}}^k}, k \in \{1,2\}$, and let $\hat{\mathcal{H}}^1$ be a given tree and $v_{i,j}^{\hat{\mathcal{H}}^1}$, $0 \leq i \leq h_1$, $1 \leq j \leq \sigma_i$ (upper index on a level i) denotes the j-th vertex on the i-th level of $\hat{\mathcal{H}}^1$, analogous to $v_{i,j}^{\hat{\mathcal{H}}^2}$ for $\hat{\mathcal{H}}^2$. Then the problem of determining the structural similarity between $\hat{\mathcal{H}}^1$ and $\hat{\mathcal{H}}^2$ is equivalent to determining the optimal alignment of

$$\begin{aligned}
S_0^{\hat{\mathcal{H}}^1} &:= w_1^{\hat{\mathcal{H}}^1}, \\
S_1^{\hat{\mathcal{H}}^1} &:= v_{1,1}^{\hat{\mathcal{H}}^1} \circ v_{1,2}^{\hat{\mathcal{H}}^1} \circ \cdots \circ v_{i,\delta_{out}(w_1^{\hat{\mathcal{H}}^1})}^{\hat{\mathcal{H}}^1}, \\
&\vdots \\
S_{h_1}^{\hat{\mathcal{H}}^1} &:= v_{h_1,1}^{\hat{\mathcal{H}}^1} \circ v_{h_1,2}^{\hat{\mathcal{H}}^1} \circ \cdots \circ v_{h_1,\sigma_{h_1}}^{\hat{\mathcal{H}}^1},
\end{aligned}$$

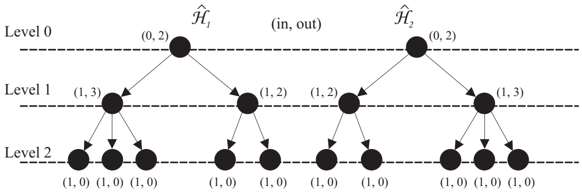

Fig. 3. Level induced outdegree and indegree sequences.

and

$$\begin{aligned}
S_0^{\hat{\mathcal{H}}^2} &:= w_2^{\hat{\mathcal{H}}^2}, \\
S_1^{\hat{\mathcal{H}}^2} &:= v_{1,1}^{\hat{\mathcal{H}}^2} \circ v_{1,2}^{\hat{\mathcal{H}}^2} \circ \cdots \circ v_{1,\delta_{out}(w_2^{\hat{\mathcal{H}}^2})}^{\hat{\mathcal{H}}^2}, \\
&\vdots \\
S_{h_2}^{\hat{\mathcal{H}}^2} &:= v_{h_2,1}^{\hat{\mathcal{H}}^2} \circ v_{h_2,2}^{\hat{\mathcal{H}}^2} \circ \cdots \circ v_{h_2,\sigma_{h_2}}^{\hat{\mathcal{H}}^2},
\end{aligned}$$

with respect to a *cost function* α (we distinguish two types of alignments: vertex-vertex, gap-vertex, vertex-gap). In order to determine the optimal alignment between two generalized trees, we define the sequences

$$S_1 := w_1^{\hat{\mathcal{H}}^1} \circ v_{1,1}^{\hat{\mathcal{H}}^1} \circ v_{1,2}^{\hat{\mathcal{H}}^1} \circ \cdots \circ v_{h_1,\sigma_{h_1}}^{\hat{\mathcal{H}}^1}, \tag{1}$$

$$S_2 := w_2^{\hat{\mathcal{H}}^2} \circ v_{1,1}^{\hat{\mathcal{H}}^2} \circ v_{1,2}^{\hat{\mathcal{H}}^2} \circ \cdots \circ v_{h_2,\sigma_{h_2}}^{\hat{\mathcal{H}}^2}, \tag{2}$$

where $S_k[i]$ denotes the i-th position of the sequence S_k and it holds $S_1[n] = v_{h_1,\sigma_{h_1}}^{\hat{\mathcal{H}}^1}, S_2[m] = v_{h_2,\sigma_{h_2}}^{\hat{\mathcal{H}}^2}$, $\mathbb{N} \ni n,m \geq 1$, $S_k[1] = w_k^{\hat{\mathcal{H}}_k}$, $k \in \{1,2\}$. Furthermore we state the definition of the alignment graph.

Definition 3.1: Let $V_{S_1,S_2} := \{(i,j)|0 \leq i \leq n, 0 \leq j \leq m\}$, $e_{Del} := (i-1,j) \to (i,j), e_{Ins} := (i,j-1) \to (i,j), e_{Subst} := (i-1,j-1) \to (i,j)$.

$$E_{S_1,S_2} := \begin{cases} e_{Del} & : i \in [1,n],\ f_{E_{S_1,S_2}}(e_{Del}) = [S_1[i], -] \\ e_{Ins} & : j \in [1,m],\ f_{E_{S_1,S_2}}(e_{Ins}) = [-, S_2[j]] \\ e_{Subst} & : i \in [1,n],\ j \in [1,m], \\ & \quad f_{E_{S_1,S_2}}(e_{Subst}) = [S_1[i], S_2[j]]. \end{cases}$$

$(i-1,j) \to (i,j)$ equals the deletion of $S_1[i]$ in S_1, $(i,j-1) \to (i,j)$ equals the insertion of $S_2[j]$ in S_1 at the i-th position, and $(i-1,j-1) \to (i,j)$ equals the substitution $S_1[i]$ to $S_2[j]$.
As an application, we consider the two trees $\hat{\mathcal{H}}_1, \hat{\mathcal{H}}_2$ represented as the sequences $S_1 := v_0 \circ v_1$ and $S_2 := v_1 \circ v_0 \circ v_1$ (simplified notation). The corresponding alignment graph is shown in figure (4). From this figure we can read off (bold edges) the alignment:

$$\begin{array}{ccc} - & v_0 & v_1 \\ v_1 & v_0 & v_1 \end{array}$$

With an edge labeling function $f_{E_{S_1,S_2}} : E_{S_1,S_2} \longrightarrow \mathbb{R}_+$ for each possible aligned pair $[a,b]$ a *cost function* $\alpha([a,b]) \in \mathbb{R}_+$ is assigned, where a, b are sequence entries of S_1 and S_2 or the gap symbol '$-$'. The algorithm with the complexity $O(|\hat{V}_1| \cdot$

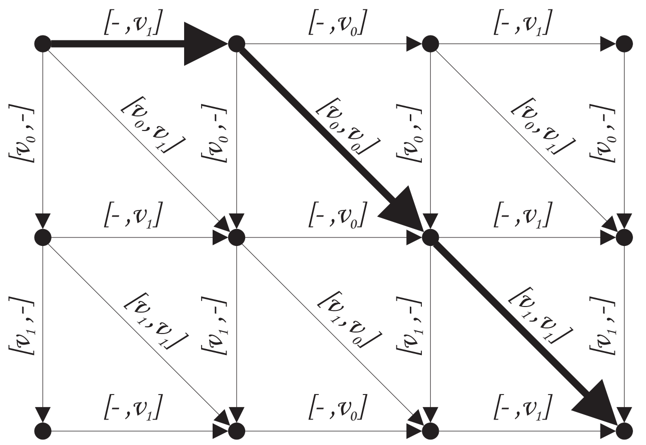

Fig. 4. Alignment graph of a sequence alignment.

$|\hat{V}_2|$) for finding the optimal alignment of the sequences generates a matrix $(\mathcal{M}(i,j))_{ij}, 0 \leq i \leq n, 0 \leq j \leq m$. Now, we find the optimal alignment on the basis of the following recursive algorithm:

$$\begin{aligned} \mathcal{M}(0,0) &:= 0, \\ \mathcal{M}(i,0) &:= \mathcal{M}(i-1,0) + \alpha(S_1[i],-) \ : \ 1 \leq i \leq n, \\ \mathcal{M}(0,j) &:= \mathcal{M}(0,j-1) + \alpha(-,S_2[j]) \ : \ 1 \leq j \leq m, \end{aligned}$$

and

$$\mathcal{M}(i,j) := \min \begin{cases} \mathcal{M}(i-1,j) + \alpha(S_1[i],-) \\ \mathcal{M}(i,j-1) + \alpha(-,S_2[j]) \\ \mathcal{M}(i-1,j-1) + \alpha(S_1[i],S_2[j]) \end{cases}$$

for $1 \leq i \leq n,\ 1 \leq j \leq m$. In order to evaluate the alignments on each level, we defined [4] the functions

$$\gamma^{out} = \gamma^{out}(i, \sigma_1^{out}, \sigma_2^{out})$$

and

$$\gamma^{in} = \gamma^{in}(i, \sigma_1^{in}, \sigma_2^{in}),$$

$\sigma_k^{out}, \sigma_k^{in} \in \mathbb{R}, k \in \{1,2\}$ in an natural way and constructed a similarity measure $d \in [0,1]$ (on the basis of these functions). $\gamma^{out}, \gamma^{in}$ are two-parametric functions, which detect the similarity of an outdegree and indegree alignment (on a level i). Finally, if we assume a set of units U and a mapping $\phi : U \times U \longrightarrow$ [0,1], we called ϕ a backward similarity measure if it satisfies the conditions

$$\phi(u,v) = \phi(v,u), \forall\, u,v \in U$$

and

$$\phi(u,u) \geq \phi(u,v), \forall\, u,v \in U.$$

Now we state the key result which has been proven in [4] for measuring the similarity for generalized trees.

Theorem 3.2: Let $\hat{\mathcal{H}}_1, \hat{\mathcal{H}}_2$, $0 \leq i \leq \rho$, $\rho := \max(h_1, h_2)$.

$$d(\hat{\mathcal{H}}_1, \hat{\mathcal{H}}_2) := \frac{\prod_{i=0}^{\rho} \gamma^{fin}(i, \sigma_1^{out}, \sigma_2^{out}, \sigma_1^{in}, \sigma_2^{in})}{\frac{\sum_{i=0}^{\rho} \gamma^{fin}(i, \sigma_1^{out}, \sigma_2^{out}, \sigma_1^{in}, \sigma_2^{in})}{\rho+1}}, \qquad (3)$$

is a backward similarity measure, where γ^{fin} is defined as

$$\begin{aligned} \gamma^{fin} &= \gamma^{fin}(i, \sigma_1^{out}, \sigma_2^{out}, \sigma_1^{in}, \sigma_2^{in}) \\ &:= \zeta \cdot \gamma^{out} + (1-\zeta) \cdot \gamma^{in}, \quad \zeta \in [0,1]. \end{aligned}$$

By construction [4] we have $d(\hat{\mathcal{H}}_1, \hat{\mathcal{H}}_2) \in [0,1]$. As a summary we note that the GTSA measures the similarity of two generalized trees by applying the technique of sequence alignments to outdegree and indegree sequences (on a level i). These alignments have both global and local significance. On the one hand, the sequence alignments will be implemented in a global sense, to compute the optimal alignment between the sequences S_1 and S_2. On the other hand, the alignments will be evaluated on the levels of the generalized trees by the function γ^{fin}. With regard to the next section we notice that our GTSA is suitable for the comparison of large generalized trees, because its complexity is enormously better then the complexity of methods which deal with isomorphic relations.

IV. LOCAL GRAPH DECOMPOSITION

In the preceding sections we introduced a method to measure the similarity between a pair of generalized trees. In this section we extend this method to construct a binary classifier for the classification of graphs. That means we construct a method which allows us to decide if two graphs are similar or not but gives us no information how similar they are. The graphs we deal with in the following are unlabeled, unweighted and undirected, hence, we simply call them graphs or networks because no special assumptions on these objects are necessary. The basic idea of this extention is a local decomposition of a graph in generalized trees. This decomposition and the construction of the binary classifier will now be described in detail.

Definition 4.1: A graph G with N nodes can be locally decomposed in a set of trees by the following algorithm:

Label all nodes from 1 to N. These labels form the label set $L_S = \{1, \ldots, N\}$. Choose a desired depth of the trees D. Choose an arbitrary label from L_S, e.g. i. The node with this label is the root node of a tree.

1) Calculate the shortest distance from node i to all other nodes in the graph G, e.g. by the algorithm of DIJKSTRA [5].
2) The nodes with distance k are the nodes in the k'th level of the tree. Select all nodes of the graph up to distance D, including the connections between the nodes. Connections to nodes with distance $> D$ are deleted.
3) Delete the label i from the label set L_S.

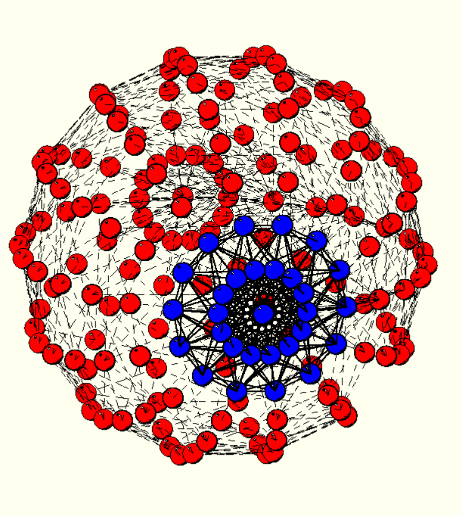

Fig. 5. A spherical graph with regular node arrangement on the surface of a sphere and regular connections between the nodes to the nearest neighbors. Shown is one local tree, resulting from the selection of the nodes up to depth $D = 2$. The root node is in the center of the two surrounding rings of nodes. (Figure was produced by Molscript [12].)

4) Repeat this procedure if L_S is not empty by choosing an arbitrary label from L_S, otherwise terminate.

This definition results in a set S_G consisting of N generalized trees of depth D. We apply to this set the GTSA introduced above and obtain a distribution of pairwise tree similarities p_{TS}.

A visualization for the extraction of one tree from a graph is given in figure 5. For didactical reasons, the nodes are regularly arranged on the surface of a sphere and the nodes are connected to its nearest neighbors. Shown are the nodes which form a tree of depth $D = 2$. The root node is in the center of the two node circles.

Suppose now we have two graphs G_1 and G_2 and we want to decide, if both graphs are similar or not or more precise are both graphs from the same class. As a solution to this problem we suggest not to compare the graphs itself as a whole but to compare local parts which can be compared in an efficient way. This decision is based on the trees similarity distributions $p_{TS}^{G_1}$ and $p_{TS}^{G_2}$ of G_1 and G_2 and the trees similarity distribution p^M which results form the union of the tree sets of the graphs, $S_M = S_{G_1} \cup S_{G_2}$. In the following we call distributions like p^M mixed and distributions like $p_{TS}^{G_1}$ or $p_{TS}^{G_2}$ pure similarity distribution. The binary classifier is based on the following definition.

Definition 4.2: Two graphs G_1 and G_2 are similar, iff the three similarity distributions $p_{TS}^{G_1}$, $p_{TS}^{G_2}$ and p^M are similar.

The idea behind this definition is a consistency check of the mutual compatibility of the contributing tree sets, because the statistics of an entity should not depend on the subset one chooses to estimate it, provided the subset is sufficient large.

Definition 4.2 maps the question of graph similarity to the similarity of distributions which can be answered much easier. In this article we will not introduce and discuss similarity measures for the comparison of distributions in depth for two reasons. First, the numerical examples given below allow a clear distinction between the obtained distributions without the application of more sophisticated methods. Second, the focus in this paper is on our novel method rather than on specialized composite approaches which could shroud our main idea.

Definition 4.2 provides a natural way to define a binary classifier for graphs.

Definition 4.3 (Binary graph classifier (BGC)): Two graphs G_1 and G_2 belong to the same class, iff the graphs are similar.

In the next section we apply this method to two classes of graphs which are artificially generated and to networks obtained from microarray experiments for different tumor stages of cervical cancer.

V. Results

We generated a set of random $\mathcal{G}_{\mathcal{RN}}$ and small world networks [21] $\mathcal{G}_{\mathcal{SW}}$ consisting each of $\#\mathcal{G}_{\mathcal{RN}} = \#\mathcal{G}_{\mathcal{SW}} = 5$ graphs. Each graph has $N = 1000$ nodes and in average $k = 10$ edges per node. The small world network was generated with a rewiring parameter $p_{rw} = 0.1$ which leads to the characteristic small world properties of short mean path lengths similar to random networks but additionally a high clustering coefficient which is low for a random network. This gives us tree sets for all graphs consisting of 1000 trees each. From each tree set we choose 20000 tree pairs randomly and calculate their similarity. The cumulative similarity distributions $P_{TS} = \int p_{TS}(x)dx$ for both classes are shown in figure 6. Here the upper curves correspond to random and the lower curves to the small world networks. The mean value for both graph classes is shown in bold line. We chose the cumulative similarity distribution because one can easily see, if the similarity measure covers the whole range or just parts from $[0, 1]$. More precisely, one can immediately read off the percentage of networks $P_{TS}(x)$ which has a similarity less or equal x.

In figure 6 one can see, that both classes have a clear characteristic which can easily distinguished. Due to the large number of trees, the fluctuations around the mean distributions are small. For random networks the fluctuations are bigger compared to the small world network. This can be explained by the construction mechanism for the small world network [21]. Initially, all nodes are arranged on a ring and the connections are regular to the next $k/2$ neighbors to each side. The small world behavior is obtained by a quite small rearrangement of the $k/2$ neighbors to one side. In our graphs it is 10%. That means the connections differ only by 10% from a regular connected ring, which makes local neighborhoods of the graph, e.g. trees, very similar to other local regions

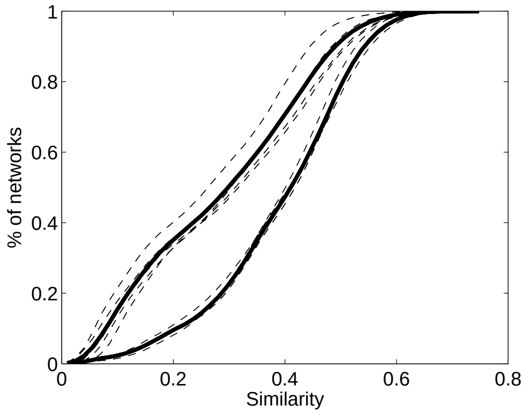

Fig. 6. Pure cumulative similarity distributions for five random (upper curves) and small world networks (lower curves). The mean value for both graph classes is shown in bold line all other curves are dashed.

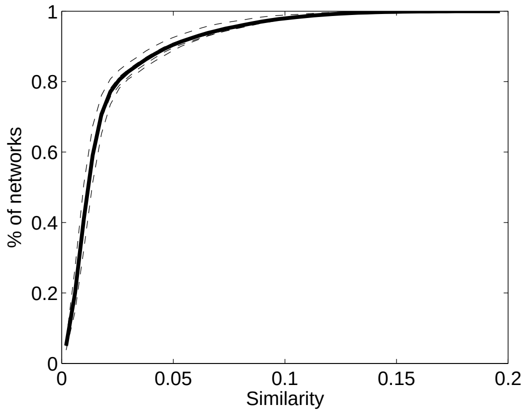

Fig. 7. Five cumulative similarity distributions for mixed tree sets $(S_{G_{RN}}, S_{G_{SW}})$. The mean value is shown in bold, all other curves are dashed.

on the same graph. This similarity is steadily reduced by increasing p_{rw}. The explanation follows now from the fact, that for $p_{rw} = 1.0$ one obtains a random network.

Now we calculate the distributions for mixed tree sets by choosing randomly 5 different pairs of tree sets $(S_{G_{RN}}, S_{G_{SW}})$. The results are shown in figure 7. One can easily see, that the resulting mixed cumulative similarity distributions are completely different to the pure similarity distributions in figure 6 and hence, random and small world networks has to be in different graph classes according to our similarity measure.

The next data set we applied our method to is from molecular biology resulting from DNA microarray experiments. We used the data from [22] which investigated the gene expression levels of different tumor stages of cervical cancer. For a summary see table II. In general, the higher the integer numbers and the letters of the tumor stages the more the cancer has grown and spread. The data include also a normal expression profile of cervical tissue indicated in table II as 'normal'. In the following we speak of the network resulting e.g. form the expression profile of tumor tissue of stage 2A, as the 2A-network. Similarly, we speak of the 2A-tree set. The networks from the expression data are obtained via a three step process [15]:

TABLE II

MICROARRAY DATA FROM [22] FOR DIFFERENT TUMOR STAGES, BASED ON THE FIGO (INTERNATIONAL FEDERATION OF GYNECOLOGISTS AND OBSTETRICS) TAGING SYSTEM, OF CERVICAL CANCER. EACH OF THE 32 (TOTAL NUMBER OF PATIENTS) ARRAYS CONTAINED 10692 GENES.

FIGO stage	Number of patients
normal	8
1B	11
2A	8
2B	5

1) Calculating the pairwise correlation coefficient for all gene profiles.
2) Prune the connections if the correlation coefficient is below a threshold Θ_{Co}.
3) Prune the connections to a node i if its clustering coefficient is below a threshold Θ_{Cl}.

These kind of networks are a prime example for the application of our method for two reasons. First, the networks are large. Typical microarray experiments involve thousands of genes which correspond to the number of nodes in the resulting network. In the data form [22] this gives networks in the order of 10^5 nodes. Second, the underlying networks are unweighted and undirected in its simplest form[1]

In figure 8 - 13 we compare the cumulative similarity distributions for all 4 different tissue samples pairwise against each other applying our binary classifier. Again, the mixed cumulative similarity distributions (shown as full line in all figures) are always different to the pure distributions except to the case in figure 9 were we compared normal- and 2A-networks.

One can interpret the results in the following way. If the representation of the expression level of different tissue conditions as undirected, unweighted graph would be appropriate we should be able to see a clear distinction between the mixed and pure distributions for all cases in figure 8 - 13, because we know that the data came from tissues of different conditions. The fact, that this is not true for the comparison of normal- and 2A-tissue suggests, that the statistics of the overall tree similarities is not sufficient to indicate the tissue's condition. This could have two reasons. First, the parameters we chose for the similarity measure 3 could be too relaxed in the sense, that different generalized tree pairs are overrated. The range difference in figure 7 compared to figure 8 - 13 for the similarity values supports this assumption. In the former

[1] This does not mean, that the underlying biomolecular process is best modeled as an unweighted or undirected graph. It means, that this kind of information is at least very difficult to extract from the data currently available.

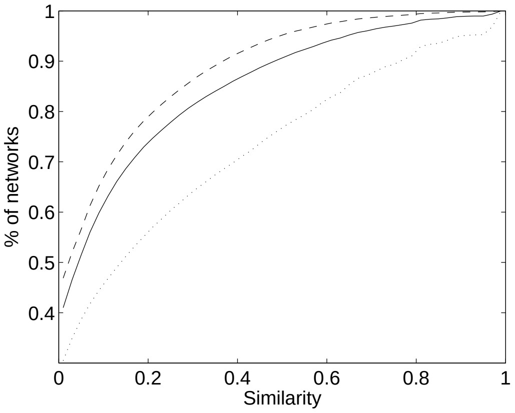

Fig. 8. Cumulative similarity distributions for the normal- (dashed line), 1B- (dotted line) and mixed- (full line) tree set.

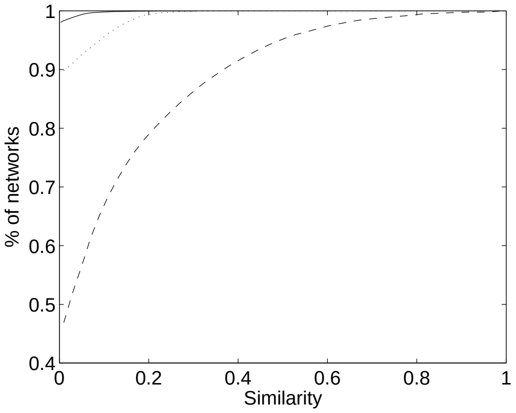

Fig. 11. Cumulative similarity distributions for the normal- (dashed line), 2B- (dotted line) and mixed- (full line) tree set.

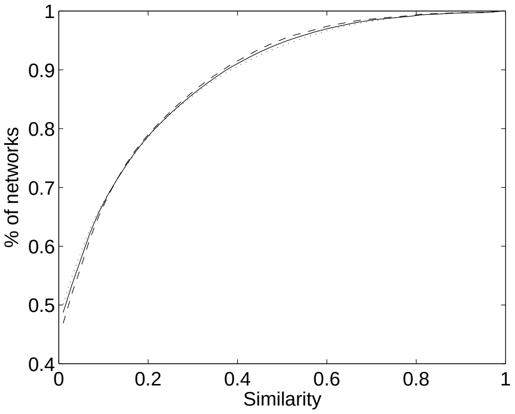

Fig. 9. Cumulative similarity distributions for the normal- (dashed line), 2A- (dotted line) and mixed- (full line) tree set.

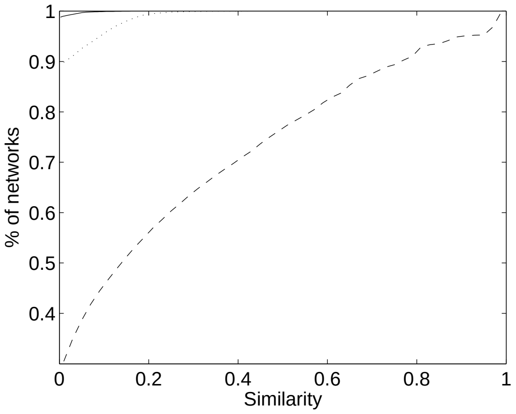

Fig. 12. Cumulative similarity distributions for the 1B- (dashed line), 2B- (dotted line) and mixed- (full line) tree set.

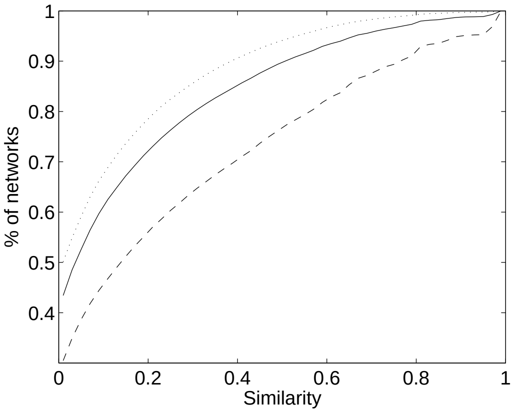

Fig. 10. Cumulative similarity distributions for the 1B- (dashed line), 2A- (dotted line) and mixed- (full line) tree set.

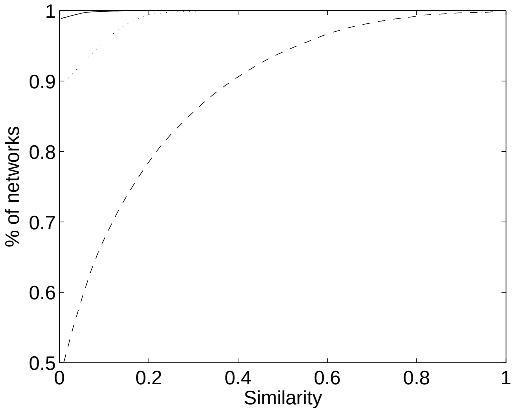

Fig. 13. Cumulative similarity distributions for the 2A- (dashed line), 2B- (dotted line) and mixed- (full line) tree set.

case the interval is $[0, 0.2]$ and in the later sill the maximal interval $[0, 1]$. It is clear, that there is no universal parameter set for similarity measure 3 which gives the best results for all network classes. Hence, the parameters seem appropriate for the comparison between random and small world networks and only partially appropriate for the differentiation for networks representing different tissue conditions. Second, the results in figure 8 - 13 are not primary induced by the parameters of equation 3 but by the genes which contribute to the network. The number of genes in all microarray experiments was 10692. It is clear, that a biased selection of genes, reducing their number by about 90%, could result in complete new network characteristics which are buried in current statistics. A short calculation shows that a subnetwork consisting of 1000 nodes in a overall network of 10000 nodes contributes only by 19% to the total number of different tree pairs one can form from this network. If one excludes the tree pairs were one tree comes from the subset and one from the rest of the network and only keeps the tree pairs from the subnetwork this number decreases further to 1%. Smaller sizes of the subnetwork continue to decrease these numbers.

VI. Conclusion

In this paper we introduced a binary graph classifier (BGC) which allows the classification of large undirected, unweighted graphs. This classifier is based essentially on the generalized tree-similarity-algorithm (GTSA) which provides a similarity measure between tree pairs by an alignment of property strings representing the trees. The application of the GTSA to undirected, unweighted graphs was enabled by a local decomposition of the graph in generalized trees resulting in a tree set on which the GTSA could be applied. The resulting similarity values for all tree pairs of the tree set determine the similarity distribution of the graph. Hence, the problem of graph classification was mapped to the classification of one dimensional similarity distributions. We demonstrated the ability to classify large graphs by the application of the BGC to two data sets consisting of graphs of size 10^3 and 10^4.

We think that the presented framework is a considerable improvement to existing graph classification approaches because we presented not only a theoretical framework suitable for small graphs but also for large graphs of sizes which are relevant for practical applications, e.g. to problems from molecular biology. Our future work, we will be guided by experimental data from microarray experiments of cancer cells to hopefully improve our method towards a reliable prediction of early tumor stages and to gain insights into the underlying complex interplay of gene expression regulation.

Acknowledgment

We would like to thank Galina Glazko and Chris Seidel for fruitful discussions.

References

[1] J. Bang-Jensen and G. Gutin, *Digraphs. Theory, Algorithms and Applications*. Springer Verlag, London-Berlin-Heidelberg, 2000

[2] R. Bellman, *Dynamic Programming*. Princeton University Press, 1957

[3] Bunke H, *Graph matching: Theoretical foundations, algorithms, and applications*. Proc. Vision Interface 2000, Montreal/Canada, 2000, 82–88

[4] M. Dehmer. and R. Gleim and A. Mehler, *A new method of measuring similarity for a special class of directed graphs*. Tatra Mountains Mathematical Publications, Slovakia, submitted for publication, August 2004

[5] E. W. Dijkstra, *A note on two problems in connection with graphs*. Numerische Math., Vol. 1, 1959, 269–271

[6] D. Gernert, *Graph grammars which generate graphs with specified properties*. Bulletin of the EATCS, Vol. 13, 1981, 13–20

[7] F Harary and E. M. Palmer, *Graphical Enumeration*. Academic Press, New York, 1973

[8] F. Kaden, *Graphmetriken und Distanzgraphen*. ZKI-Informationen, Akad. Wiss. DDR, Vol. 2 (82), 1982, 1–63

[9] F. Kaden, *Graph metrics and distance-graphs*. In: Graphs and other Combinatorial Topics, ed. M. Fiedler, Teubner Texte zur Math., Leipzig, Vol. 59, 1983, 145–158

[10] F. Kaden, *Graphmetriken und Isometrieprobleme zugehöriger Distanzgraphen*. ZKI-Informationen, Akad. Wiss. DDR, 1986, 1–100

[11] I. Koch and T. Lengauer and E. Wanke, *An algorithm for finding maximal common subtopologies in a set of protein structures*. Journal of Computational Biology, Vol. 3, (2), 1996, 289–306

[12] P. J. Kraulis, *Molscript: A Program to Produce Both detailed and schematic plots of protein structures*. Journal of Applied Crystallography, Vol. 24, 1991, 946–950

[13] A. Mehler, M. Dehmer, R. Gleim: *Towards logical hypertext structure. A graph-theoretic perspective*, Proc. of I2CS'04, Guadalajara/Mexico, Lecture Notes in Computer Science, Berlin-New York: Springer, 2004

[14] R. C. Read and D. G. Corneil, *The graph isomorphism disease*. Journal of Graph Theory, Vol. 1, 1977, 339–363

[15] J. Rougemont and P. Hingamp, *DNA microarray data and contextual analysis of correlation graphs*. BMC Bioinformatics, Vol. 4, 2003, 4–15

[16] L. G. Shapiro, *Organization of relational models*. Proc. Intern. Conf. on Pattern Recognition, München, 1982, 360–365

[17] M. I. Skvortsova and I. I. Baskin and I. V. Stankevich and V. A. Palyulin and N. S. Zefirov, *Molecular similarity in structure-property relationship studies. Analytical description of the complete set of graph similarity measures*. International symposium CACR-96, 1996

[18] F. Sobik, *Graphmetriken und Klassifikation strukturierter Objekte*. ZKI-Informationen, Akad. Wiss. DDR, Vol. 2 (82), 1982, 63–122

[19] F. Sobik, *Graphmetriken und Charakterisierung von Graphklassen*. 27. Internat. Wiss. Koll., TH-Ilmenau, Vol. 2 (82), 1982, 63–122

[20] J. R. Ullman, *An algorithm for subgraph isomorphism*. J. ACM, Vol. 23 (1), 1976, 31–42

[21] D. J. Watts and S. H. Strogatz *Collective dynamics of 'small-world' networks*. Nature, Vol. 393, 1998, 440–42

[22] Y. F. Wong et.al. *Expression Genomics of Cervical Cancer: Molecular Classification and Prediction of Radiotherapy Response by DNA Microarray*. Clinical Cancer Research, Vol. 9, 2003, 5486–5492

[23] B. Zelinka, *On a certain distance between isomorphism classes of graphs*. Časopis pro p̆est. Mathematiky, Vol. 100, 1975, 371–373

[24] K. Zhang and R. Statman and D. Shasha, *On the editing distance between unordered labeled trees*. Int. Inf. Process. Lett., Vol. 42 (3), 1992, 133–139

[25] K. Zhang and J. Wang and T. L. Jason and D. Shasha, *On the editing distance between undirected acyclic graphs*. Int. J. Found. Comput. Sci., Vol. 7 (1), 1996, 43–57