Perspectives of Graph Mining Techniques

Matthias Dehmer[1], Frank Emmert-Streib[2], Olaf Wolkenhauer[1]

[1] University of Rostock,
Department of Computer Science,
Systems Biology and Bioinformatics,
Albert Einstein Str. 21,
18051 Rostock, Germany,
{dehmer|ow}@informatik.uni-rostock.de
[2] Stowers Institute for Medical Research,
1000 E. 50th Street, Kansas City, MO 64110, USA,
fes@stowers-institute.org

Abstract. In this paper we review known approaches to measure the structural similarity of graphs and compare them with a novel similarity measure recently introduced [8]. The major advantage of this new similarity measure is that it can be efficiently computed in contrast to the classical measures, because it is based on the optimal alignment of strings which can be found by dynamic programming. Further, we present some applications of this new graph similarity measure to problems from web structure mining and molecular biology that are out of range for the classical similarity measures because their computational complexity is intractable for these cases.

1 Introduction

In the last few years *Graph Mining* techniques [14] became very popular. Major application areas of Graph Mining Techniques are currently Biology, Chemistry, and Computer Science [14, 21]. Current problems in Graph Mining are determining the structural similarity between graphs [15, 16, 19, 26], exploration of certain sets of vertices in a graph, e.g., in terms of vertex centrality [12], finding frequent subgraph patterns in large sparse graphs [14, 18, 24], and determining certain edge sets in a graph, e.g., shortest paths [12]. In this paper we will focus on perspectives of the graph similarity concept in two current application areas: Web Structure Mining [7] and Computational Biology [23]. More precisely, our goal is to present known and novel approaches for measuring the structural similarity of graphs in order to classify and analyze graph corpora, respectively.

The structural comparison of graphs is a difficult and outstanding problem. Most of the classical contributions dealing with graph similarity measures are based on isomorphic and subgraph relations [15, 22]. An example of such a graph similarity measure is the well-known Zelinka-distance [26].The Zelinka-distance is based on the principle that two graphs are more similar, the bigger the common induced isomorphic subgraph is. Zelinka was the first who introduced this measure for unlabeled graphs. Sobik [19, 20] and Kaden [15] generalized this measure for arbitrary graphs including also labeled

graphs of different orders (number of vertices). Because of the well known fact, that the subgraph isomorphism problem is $\mathcal{NP}$-complete [22] the measures mentioned above are not suitable for the application to graphs of large order. In this paper we review some classical graph similarity methods, e.g., isomorphic relations and *graph edit distance* models [4–6], as well as our our novel approach to measure the structural similarity of a special graph class introduced by DEHMER [8]. We call this new graph class *generalized trees* [9, 17], because this graph class includes ordinary trees [1] as a special case. Further, we present some novel applications of the new graph similarity measure [8] to problems from web structure mining [8] and molecular biology.

This paper is organized in the following way: In order to distinguish the classical methods from our novel method [8] we review in section (2) some classical graph similarity measures. In section (3) we present the construction of our graph similarity measures. Applications will be presented in section (4). This paper finishes in section (5) with conclusions and an outlook.

2 Structural Similarity Measures of Graphs

In this section, we review some classical graph similarity measures. These approaches can be mainly divided into the following four categories.

2.1 Isomorphic Measures

Measures based on isomorphic relations, e.g., [15, 16, 19, 20, 26]: Before we present the graph similarity measures based on subgraph isomorphic relations, we need some graph theoretical preliminaries. Let $\mathcal{H} := (V, E)$, $E \subseteq V \times V$ be an unlabeled, directed graph. $\mathcal{G} := (\hat{V}, \hat{E})$ with $\hat{V} \subseteq V$ and $\hat{E} \subseteq E$ denotes the partial graph of $\mathcal{H}$, $\mathcal{G} \subseteq \mathcal{H}$. If $\hat{E} = E \cap (\hat{V} \times \hat{V})$ hold, then we call $\mathcal{G}$ the induced subgraph of $\mathcal{H}$. Further, we give the definition of graph isomorphism that provides the structural equivalence of two graphs [20]. Let $\mathcal{H} := (V, E)$ and $\mathcal{G} := (\hat{V}, \hat{E})$ be directed graphs. We call $\mathcal{H}$ and $\mathcal{G}$ isomorphic ($\mathcal{H} \cong \mathcal{G}$) if it exists a bijective mapping $\phi : V \longrightarrow \hat{V}$ such that

$$E \ni (v_i, v_i) :\Longleftrightarrow (\phi(v_i), \phi(v_j)) \in \hat{E}. \tag{1}$$

The mapping ϕ is called the isomorphism of $\mathcal{H}$ on $\mathcal{G}$.

The key result for graph similarity measuring due to ZELINKA [26] is the following theorem.

Theorem 1. *Let $\mathcal{H}$, $\tilde{\mathcal{H}}$ be an unlabeled graphs without loops and multiple edges. Further, let $|V| = |\tilde{V}| = n$. $\overline{SUB}_m(\mathcal{H})$ denotes the set of induced subgraphs of order m. $\mathcal{H}^\star$ denotes the isomorphism classes of such graphs in which $\mathcal{H}$ lies and let*

$$SUB_m(\mathcal{H}) := \{\mathcal{H}^\star \,|\, \mathcal{H} \in \overline{SUB}_m(\mathcal{H})\}. \tag{2}$$

$SUB_m(\mathcal{H})$ is just the set of isomorphism classes in which the induced subgraphs of $\mathcal{H}$ lie with order m. Then,

$$d_Z(\mathcal{H}, \tilde{\mathcal{H}}) := n - SIM(\mathcal{H}, \tilde{\mathcal{H}}), \tag{3}$$

is a graph metric, where

$$SIM(\mathcal{H}, \tilde{\mathcal{H}}) := \max\{m | SUB_m(\mathcal{H}) \cap SUB_m(\tilde{\mathcal{H}}) \neq \emptyset \tag{4}$$

hold.

A generalization of this measure to arbitrary, also labeled graphs of different orders was given by Sobik [19] and Kaden [15].

Theorem 2. *Let $\mathcal{H} := (V, E, f_V, f_E, A_V, A_E)$ be a finite, labeled and directed graph. A_V, A_E denote finite, non-empty vertex and edge alphabets and $f_V : V \to A_V$, $f_E : E \to A_E$ the associated vertex and edge labeling functions. Now, let $\mathcal{H}$ and $\tilde{\mathcal{H}}$ be finite, labeled graphs of arbitrary orders. Then,*

$$d_S(\mathcal{H}, \tilde{\mathcal{H}}) := \max\{|\mathcal{H}|, |\tilde{\mathcal{H}}|\} - SIM(\mathcal{H}, \tilde{\mathcal{H}}) \tag{5}$$

is a graph metric.

As mentioned before, these measures are not applicable for practical graph similarity problems if these graphs have large orders, because the subgraph isomorphism problem is $\mathcal{NP}$-Complete. Hence, one can not obtain algorithms with polynomial time complexity.

2.2 Measures based on graph transformations

Measures based on graph transformations are from, e.g., [4–6]. These graph similarity measures [6, 27] are mostly based on the well known graph edit distance [6]. The graph edit distance is based on weighted transformation steps, e.g., deletions, substitutions, and insertions of vertices and edges. Since there is an infinite number of different possibilities to transform one graph into another graph, the similarity of the graphs is defined as the minimum cost of transformations. Before we present a key result due to Bunke et al. [4] we first give the definition of an *Optimal Inexact Match* [4]. Optimal Inexact Match is a sequence of transformations (insertions, deletions and substitutions of vertices) which transforms $\mathcal{H}_1$ to $\mathcal{H}_2$ by producing minimal transformation costs. Assuming $m_1, m_2, \ldots, m_n$ are all possible Inexact Matches between $\mathcal{H}_1$ and $\mathcal{H}_2$. Then, the Optimal Inexact Match m' is defined by $c(m') = \min\{c(m_i) | \, 1 \leq i \leq n\}$, where $c(m_i)$ denotes the costs of m_i. Finally, we give the theorem of Bunke et al. [4] based on the graph edit distance.

Theorem 3. *Let $d(\mathcal{H}_1, \mathcal{H}_2)$ be the costs for determining the Optimal Inexact Match between $\mathcal{H}_1$ and $\mathcal{H}_2$. $d(\mathcal{H}_1, \mathcal{H}_2)$ is a graph metric.*

For general graphs, there is no efficient algorithm to determine the graph edit distance as noticed by Zhang et al. [27].

2.3 Graph Grammars

Measures based on graph grammars are from, e.g., [10, 11]. For example, Gernert [10, 11] stated a known approach within the inexact graph matching paradigm [6] based on

graph grammars [10]. In order to give the key result of GERNERT [10, 11], we consider a set of graphs $S := \{G_1, G_2, \ldots, G_n\}$, where G_i, $1 \leq i \leq n$ are assumed to be connected. Furthermore we assume a graph grammar α, that generates a set $\tilde{S} := \{\tilde{G}_1, \tilde{G}_2, \ldots, \tilde{G}_p\}$, $S \subseteq \tilde{S}$. Based on the function

$$f(\tilde{G}_i, \tilde{G}_k) := \begin{cases} 0 & : \tilde{G}_i \text{ isomorphic to } \tilde{G}_k \\ 1 & : \tilde{G}_i \longrightarrow \tilde{G}_k \text{ one replacement step} \\ \text{undefined} & : \text{else,} \end{cases}$$

GERNERT defined so called path lengths between $\tilde{G}_r$ and $\tilde{G}_s$, if $\tilde{G}_s$ can be generated by a finite sequence of steps, starting from $\tilde{G}_r$. Further it exists at least one graph, e.g., the start graph, $G^\star$ with $G^\star \longrightarrow \tilde{G}_r$ and $G^\star \longrightarrow \tilde{G}_s$. If we now denote the path length with l we obtain [10, 11]

Theorem 4. *$d(G_i, G_k)$ defined by*

$$d(G_i, G_k) := \min \left\{ l(G^\star, G_i) + l(G^\star, G_k) \,\middle|\, G^\star \in \tilde{S} \wedge G^\star \subseteq G_i \wedge G^\star \subseteq G_k \right\}$$

is a graph metric.

The application of graph similarity measures based on graph grammars is complex, because the underlying grammar is difficult to define.

2.4 Graph Kernels

Measures based on graph kernels are from, e.g., [**?**,13]. Here, we outline the supervised learning approach of HORVÁTH et al. [13] for determining an efficient graph kernel based on a given graph class. Generally, a graph kernel [**?**] is a function $k : \mathcal{G} \times \mathcal{G} \longrightarrow \mathbb{R}$, that detects the similarity between training examples of a graph set $\mathcal{G}$. Most graph kernels are based on the principle that one first has to determine the frequency of subgraph patterns of the given graph set and then apply the characteristic property of the kernel function, e.g. a tree property, to the obtained subgraphs. Following these procedure HORVÁTH et al. proposed a graph kernel [13], that is based on cyclic and tree patterns of the underlying graphs. The definition of these kernel function is based on the two following steps:

1. Divide the graph objects into certain sets of characteristic graph patterns, e.g., trees.
2. Determine the cut-set of pattern sets.

Starting from these steps, HORVÁTH et al. defined the kernel function (CP=Cyclic Patterns) as

$$k_{CP}(G_i, G_j) := |C(G_i) \cap C(G_j)| + |\tau(G_i) \cap \tau(G_j)|$$

where $C(G)$ and $\tau(G)$ denotes the set of cyclic and tree patterns of G, respectively. However, HORVÁTH et al. [13] showed that the computation of $k_{CP}(G_i, G_j)$ requires exponential time complexity. In order to compute the graph kernel efficiently, HORVÁTH et al. set constrains, like *simple cycles* [13] and proved efficient upper bounds for the number of simple cycles of a graph G. We remark that graph kernels based on (i) decomposition of the given graph and (ii) determining cut sets of certain graph subsets are generally difficult to obtain. Furthermore, algorithms, computing the kernel function efficiently, are not obvious available.

3 Structural Similarity Measures of Generalized Trees

In this section we present the construction of our new method to measure the structural similarity of unlabeled generalized trees which were first introduced in [17]. Generalized trees are defined as follows [8, 17].

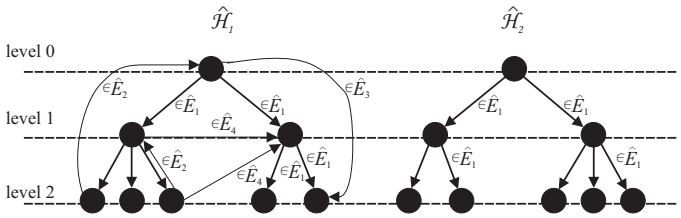

Fig. 1. $\hat{\mathcal{H}}_1$ shows a generalized tree. His edge types satisfy Definition (1). $\hat{\mathcal{H}}_2$ shows a ordinary directed rooted tree, which consists only of (bold) edges $e \in \hat{E}_1$. An edge $e \in \hat{E}_1$ over-jumps always just one level, $e \in \hat{E}_3$ over jumps at least one level, $e \in \hat{E}_4$ does not necessarily over-jump a level.

Definition 1. *Let $\mathcal{H} = (\hat{V}, E_1)$ be an directed, rooted tree. Let $m_{\hat{V}} : \hat{V} \longrightarrow A_{\hat{V}}$ be a vertex labeling function and $A_{\hat{V}}$ denote a vertex alphabet. The vertex set $\hat{V}$ is defined by*

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

where $v_{i,j}$ denotes the j-th vertex on the i-th level, $0 \leq i \leq h$, $1 \leq j \leq \sigma_i$. h denotes the depth of $\mathcal{H}$ and σ_i is the number of vertices on level i. The edge set $\hat{E} := \hat{E}_1 \cup \hat{E}_2 \cup \hat{E}_3 \cup \hat{E}_4$ is defined as [17]*:*

- *($\hat{E}_1$) is the edge set of the underlying directed rooted tree $\mathcal{H}$.*
- *($\hat{E}_2$) Up-edges associate analogously nodes of the tree hierarchy with one of their (dominating) predecessor nodes.*
- *($\hat{E}_3$) Down-edges associate nodes of the tree hierarchy with one of their (dominated) successor nodes in terms of that tree hierarchy.*
- *($\hat{E}_4$) Cross-edges associate nodes of the tree hierarchy, none of which is an (immediate) predecessor of the other in terms of the tree hierarchy.*

If $\hat{E}_2$, $\hat{E}_3$ and $\hat{E}_4 \neq \emptyset$ then $\hat{\mathcal{H}} = (\hat{V}, \hat{E}, m_{\hat{V}}, A_{\hat{V}})$ denotes a generalized tree. If we set $A_{\hat{V}} = \emptyset$, the generalized tree is unlabeled.

Figure (1) shows an example of a generalized tree. One can easily see the difference to an ordinary directed rooted tree. From the definition (1) it follows that because of the additional edge-types [17], Up-edges, Down-edges and Cross-edges, generalized trees have a more complex structure than ordinary directed, rooted trees. Hence, this graph class can represent more complex problems including all problems that can be represented by ordinary trees.

In the following we give just an outline of the main construction steps of our structural similarity measuring for generalized trees:

1. Transform the generalized trees in linear structures we call property strings.
2. Derive similarity scores from the alignments of the property strings in order to measure the structural similarity of generalized trees.

This means, we transform a graph similarity problem into a string similarity problem to develop an efficient graph similarity measures. In more detail, the main idea of our similarity measure is based on the derivation of property strings for each generalized tree and then to align the property strings by a dynamic programming technique [2]. We call these strings property strings, because their components represent structural properties of the generalized trees. From the resulting alignment we obtain a value of the scoring function which is minimized during the alignment process. The similarity of two generalized trees is then given as the cumulation of local similarity functions which weight two alignment types: out-degree and in-degree alignments on a generalized tree level. Now, let $\hat{\mathcal{H}}^1$ and $\hat{\mathcal{H}}^2$ be generalized trees satisfying Definition (1). Then the problem to determine the structural similarity between $\hat{\mathcal{H}}^1$ and $\hat{\mathcal{H}}^2$ is equivalent the to find the optimal alignment of the property stings. The key result is given in the following theorem of Dehmer [8].

Theorem 5. *Let $\hat{\mathcal{H}}^1, \hat{\mathcal{H}}^2$ be generalized trees, $0 \leq i \leq \rho$, $\rho := \max(h_1, h_2)$.*

$$d_1(\hat{\mathcal{H}}^1, \hat{\mathcal{H}}^2) := \frac{\sum_{i=0}^{\rho} \lambda_i \cdot \gamma^{fin}(i)}{\sum_{i=0}^{\rho} \lambda_i}, \tag{6}$$

$$d_2(\hat{\mathcal{H}}^1, \hat{\mathcal{H}}^2) := \frac{\sum_{i=0}^{\rho} \gamma^{fin}(i)}{\rho + 1}, \tag{7}$$

$$d_3(\hat{\mathcal{H}}^l 1, \hat{\mathcal{H}}^2) := \frac{\prod_{i=0}^{\rho} \gamma^{fin}(i)}{d_2(\hat{\mathcal{H}}^1, \hat{\mathcal{H}}^2)}, \tag{8}$$

is a family $(d_i(\hat{\mathcal{H}}^1, \hat{\mathcal{H}}^2))_{1 \leq i \leq 3}$ of Backward similarity measures, where $\gamma^{fin}(i)$ is the weighted sum of the in- and out-degree alignments. Further it holds that $(d_i(\hat{\mathcal{H}}^1, \hat{\mathcal{H}}^2))_{1 \leq i \leq 3} \in [0, 1]$.

We just note that this result can be generalized to labeled generalized trees [8].

4 Known and Novel Applications of Graph Mining Techniques

We present in this section some applications of the method from section (3) in web structure mining and molecular biology. In order to show, that there is a demand for new graph-theoretical methods for the analysis of hypertext structures we first present in section (4.1) simple graph-theoretical measures which describes properties of hypertext structures and then present applications of our measure.

4.1 Known Graph-Theoretical Applications in Hypertext Analysis

In the early days of hypertext structure modeling many graph-theoretic measures for the analysis of hypertexts were developed, e.g., [3, 25]. In the following we will see, that those existing simple measures are not suitable for a similarity-based structural analysis of hypertext graphs, because they can not capture enough structural information of the

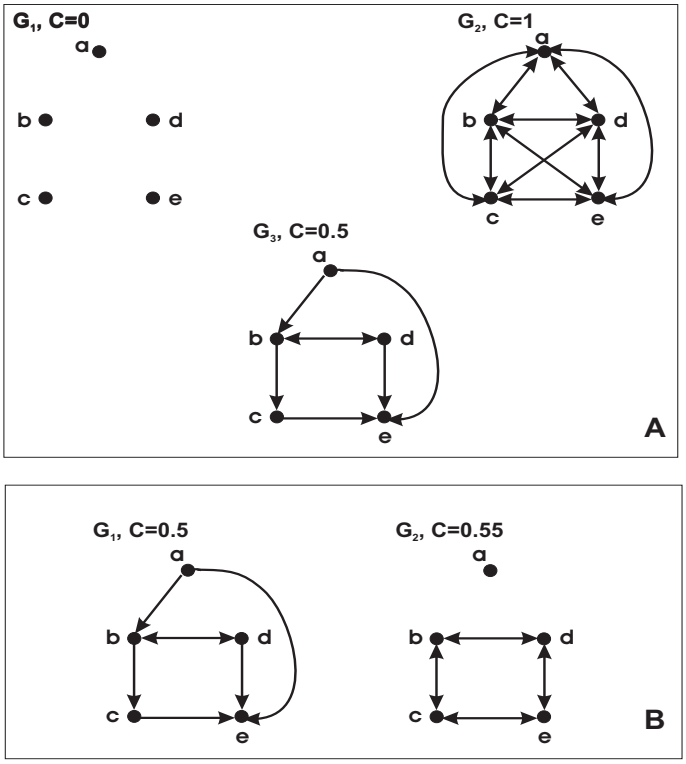

Fig. 2. (A) G_1 is the empty graph with five vertices. G_2 is the complete graph with five vertices. G_3 has also five vertices with $C = 0.5$. (B) Two Graphs G_1 and G_2 with similar compactness values. Notice, that G_2 has one isolated vertex.

underlying graphs. In the field of structural analysis of hypertexts there are many measures that are called *indices* describing structural properties of hypertexts [3, 25]. The characteristic property of an index is that the described structural property, e.g., connectedness, is mapped to a normalized value. For example, BOTAFOGO et al. [3] defined the well known index *Compactness* of a directed hypertext graph $\mathcal{H}$ as

$$C := \frac{(|V|^2 - |V|) \cdot \mathcal{K} - \sum_{i=1}^{|V|} \sum_{j=1}^{|V|} c_{ij}}{(|V|^2 - |V|) \cdot \mathcal{K} - (|V|^2 - |V|)} \in [0, 1]. \tag{9}$$

C measures the connectedness of a hypertext graph.

$$(c_{ij})_{ij} := \begin{cases} w_{ij} : \text{if } w_{ij} \text{ exists} \\ \mathcal{K} \quad : \text{else,} \end{cases} \tag{10}$$

denotes the converted distance matrix and w_{ij} denotes the shortest path from v_i to v_j. $\mathcal{K}$ defines the *conversion constant* [3], $|V|$ denotes the number of vertices of $\mathcal{H}$ and BOTAFOGO et al. [3] set $\mathcal{K} = |V|$. From the definition of C we can derive that $C = 1$ iff $\mathcal{H}$ is completely connected and $C = 0$ iff $\mathcal{H} = (V, \emptyset)$. As an example, Figure (2) (A) shows exemplarily the graphs G_1, G_2 and G_3. It holds $C_{G_1} = 0$, $C_{G_2} = 1$ and $C_{G_3} = 0.5$. Now, suppose for two hypertext graphs $\mathcal{H}_1$, $\mathcal{H}_2$ holds $C_1 \approx C_2$. From the definition of C it is clear that the graph structures can be different. Therefore, the index C is not suitable for determining intervals which contain similar hypertext graphs in terms for deriving quality features, e.g., "positive navigation behavior" (e.g., see Figure (2) (B)).

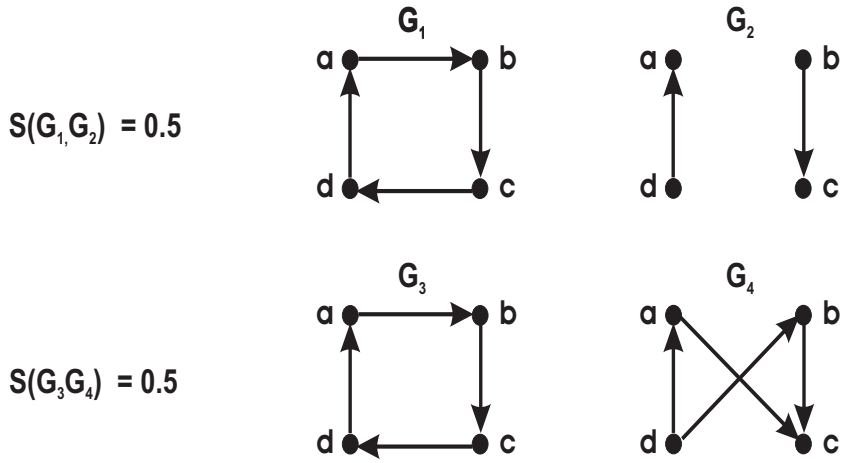

Fig. 3. Two Graphs with equal values of S. Notice, that G_1 and G_2 have different edge structures.

In terms of graph similarity we consider the simple index

$$S(G_1, G_2) := \frac{|E_1| \cap |E_2|}{\max\left(|E_1|, |E_2|\right)} \in [0, 1].$$

We notice that S is simply based of the underlying edge cut sets. From Figure (3) we see that S does not capture enough structural information of the graph. Hence, S is not a meaningful index for measuring the structural similarity of hypertext graphs. A further index is *Multiplicity* defined by Winne et al. [25]. Multiplicity is defined as the ratio of the edge cut set of two graphs to the number of all possible edges. In terms of expressiveness we have the same situation as mentioned above.

4.2 Novel Applications in Web Structure Mining

Based on the approach stated in section (3), we propose the analysis of graph-based hypertext structures based on the following steps:

1. We represent web-sites or web-pages in a corpus as generalized trees [17]. Based on the parametric model described in section (3), we are able to emphasize different structure types of generalized trees during the evaluation of the alignment. That means, if we set $\zeta = 1$, we treat a document structure as a directed rooted tree. That means we align solely the out-degree property strings induced by edges $e \in \hat{E_1}$. In contrast to this, if we set $\zeta = 0$, then we align the property strings induced by in-degree sequences only. In the most application cases we set $\zeta = \frac{1}{2}$, that means we weigh the out-degree and in-degree property strings equally.
2. Create the similarity matrix by computing all pairs of similarity values based on the underlying graph corpus. Then we can depict the so called cumulative similarity distribution. That means, if we consider the corresponding two-dimensional plot, then the X-axis denotes the similarity values of and the Y-axis denotes the percentage rate of generalized trees which have a similarity value (from the X-axis). Based on this distribution we can answer the important question how structurally different the graphs of the corpus are [8]. Finally, the determination of the cumulative similarity

distribution based on a corpus leads us to a further applications in Web Structure Mining and to a better understanding of web-based hypertext structures and their interactions.
3. Starting from the computed matrices we can apply multivariate data analysis methods, e.g. clustering techniques. A direct application of the stated procedure is the task of structural filtering [8] of web-based documents.

4.3 Novel Applications in Molecular Biology

To be useful for problems form molecular biology the similarity measure for generalized trees presented in section (3) was extended by Emmert-Streib et al. to unweighted, undirected graphs [9]. The unweighted, undirected graphs were obtained from microarray data of cervical cancer and different graphs represented different tumor stages. That means, the classification problem of tumor stages was mapped to a classification problem of graphs. To solve the graph classification problem each graph was locally decomposed in a generalized tree where the hierarchy of the generalized tree is induced naturally by the Dijkstra distance. Performing this decomposition for each node in a graph results in a set of generalized trees for each graph. Finally, the sets of generalized trees were classified by application of the method introduced in section (3). It was demonstrated that this new graph classification method (binary similarity measure) can classify graphs of order 10^4 and, hence, tumor stages of cervical cancer successfully [9]. The high order of the graphs comes form the large number of genes that can simultaneously be measured by microarray experiments and is already for lower organisms like yeast about 6000. Graphs of this order are not investigatable by the classical graph similarity measures, because of their intractable complexity. This example demonstrates clearly, that the new graph classification method is useful for a practical application to problems in molecular biology.

5 Conclusions and Outlook

In this paper we reviewed known approaches to measure the structural similarity of graphs. Most approaches we have presented are useful for theoretical investigations, but not for practical applications. Further, we reviewed the novel approach [8] to measure the structural similarity of generalized trees. As shown in section (3), we obtained a family of similarity measures rather than a single measure. In the future we plan to analyze the graph class of generalized trees in deepth to derive properties that might be useful for several areas of application, e.g., web structure mining or molecular biology.

References

1. J. Bang-Jensen, G. Gutin, *Digraphs. Theory, Algorithms and Applications.* Springer Verlag, London-Berlin-Heidelberg, 2000.
2. R. Bellman, *Dynamic Programming.* Princeton University Press, 1957.
3. R. A. Botafogo, B. Shneiderman: *Structural analysis of hypertexts: Identifying hierarchies and useful metrics*, ACM Trans. Inf. Syst. 10 (2), 1992, 142-180
4. H. Bunke, G. Allermann: *A metric on graphs for structural pattern recognition.* In: SIGNAL PROCESSING II: Theorie and Applications. Editor: H. W. Schüssler, 1983, 257–260.

5. H. Bunke, *Graph matching: Theoretical foundations, algorithms, and applications.* Proc. Vision Interface 2000, Montreal/Canada, 2000, 82–88.
6. H. Bunke: *Recent developments in graph matching.* Proc. of the 15-th Int. Conf. on Pattern Recognition, Vol. 2, Barcelona/Spain, 2000, 117–124.
7. S. Chakrabarti: *Mining the Web. Discovering Knowledge from Hypertext Data*, Morgen and Kaufmann Publishers, 2003
8. M. Dehmer: *Strukturelle Analyse web-basierter Dokumente*, Deutscher Universitäts-Verlag, Gabler Edition Wissenschaft, Serie: Multimedia und Telekooperation, Editors: Lehner F., Bodendorf F., Februar 2006
9. F. Emmert-Streib, M. Dehmer, J. Kilian: *Classification of large graphs by a local tree decomposition.* Proceedings of the 2005 International Conference on Data Mining (DMIN'05), Editors: H.R. Arabnia, A. Scime (2005) 200–207.
10. D. Gernert: *Measuring the similarity of complex structures by means of graph grammars*, Bulletin of the EATCS, Vol. 7, 1979, 3–9
11. D. Gernert, *Graph grammars which generate graphs with specified properties.* Bulletin of the EATCS, Vol. 13, 1981, 13–20
12. F. Harary, *Structural models. An introduction to the theory of directed graphs*, Wiley, New York, 1965
13. Horváth T., Gärtner T., Wrobel S.: *Cyclic pattern kernels for predictive graph mining*, Proceedings of the 2004 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2004, 158-167
14. A. Inokuchi, T. Washio, H. Motada, *Complete Mining of frequent patterns from graphs: Mining graph data*, Machine Learning, Vol. 50 (3), 2003, 321–354
15. F. Kaden: *Graphmetriken und Distanzgraphen.* ZKI-Informationen, Akad. Wiss. DDR, 1982, 1–63.
16. I. Koch, T. Lengauer, 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.
17. A. Mehler, M. Dehmer, R. Gleim, *Towards logical hypertext structure. A graph-theoretic perspective*, Proceedings of I2CS'04, Guadalajara/Mexico, Lecture Notes in Computer Science, Berlin-New York: Springer, 2004.
18. U. Rückert, S. Kramer, *Frequent free tree disvovery in graph data.* Proceedings of the 2004 ACM Symposium on Applied Computing, 2004, 564–57.
19. F. Sobik, *Graphmetriken und Klassifikation strukturierter Objekte.* ZKI-Informationen, Akad. Wiss. DDR, Vol. 2 (82), 1982, 63–122.
20. F. Sobik, *Modellierung von Vergleichsprozessen auf der Grundlage von Ähnlichkeitsmaßen für Graphen.* ZKI-Informationen, Akad. Wiss. DDR, Vol. 4, 1986, 104–144.
21. N. Trinajstic, *Chemical Graph Theory.* CRC Press, 1992.
22. J. R. Ullman, *An algorithm for subgraph isomorphism.* J. ACM, Vol. 23 (1), 1976, 31–42.
23. J T. L. Wang, C. H. Wu, P. P. Wang : *Computational Biology and Genome Informatics*, World Scientific Publishing Company, 2003
24. T. Washio, H. Motoda: *State of the art of graph-based data mining.* ACM SIGKDD Explorations Newsletter, Vol. 5(1), 2003, 59–68.
25. P. H. Winne., L. Gupta, J. C. Nesbit: *Exploring individual differences in studying strategies using graph theoretic statistics*, The Alberta Journal of Educational Research, Vol. 40, 1994, 177–193
26. B. Zelinka, *On a certain distance between isomorphism classes of graphs.* Časopis pro pest. Mathematiky, Vol. 100, 1975, 371–373.
27. K. Zhang, J. Wang, T. L. Jason, D. Shasha: *On the editing distance between undirected acyclic graphs*, Int. J. Found. Comput. Sci., Vol. 7(1), 1996, 43–57