Feature-Based Graph Similarity with Co-Occurrence Histograms and the Earth Mover's Distance

Marc Wichterich, Anca Maria Ivanescu, Thomas Seidl
Data Management and Exploration, RWTH Aachen University
{wichterich, ivanescu, seidl}@cs.rwth-aachen.de

Abstract: Graph structures are utilized to represent a wide range of objects including naturally graph-like objects such as molecules and derived graph structures such as connectivity graphs for region-based image retrieval. This paper proposes to extend the applicability of the Earth Mover's Distance [RTG98] (EMD) to graph objects by deriving a similarity model with a representation of structural graph features that is compatible with the feature signatures of the EMD. The aim is to support the search for a graph in a database from which the query graph may have originated through limited structural modification. Such query graphs with missing or additional vertices or edges may be the result of natural processes of decay or mutation or may stem from measuring methods that are inherently error-prone, to name a few examples.

1 Introduction and Related Work

Graphs are widely used data structures for modeling complex objects. For example, in computer vision and pattern recognition graphs are extracted from complex objects, stored in databases, and are used for graph-based shape recognition [SKK04] or for object recognition in general [HHEW04]. In the biomedical field, Takahashi investigates the structural similarity of chemicals with similar biological activity by using graphs to represent the structure of the chemicals [Tak04]. These applications exemplify the need for graph similarity measures that allow for the clustering of graphs in a database, or for finding graphs in a graph database that are similar to a query graph.

In the absence of a canonical representation of graphs, deciding if two graphs are isomorph (i.e., identical but for a renaming of the vertices) is a computationally expensive task. Its generalization, the subgraph isomorphism problem, is known to be NP-complete. When attempting to find all graphs in a database that contain a subgraph that is isomorph to some query graph, it is possible to use lower-bounding filtering techniques to quickly rule out some candidates and refine the rest with the computationally expensive exact matching. For example, the GraphGrep approach indexes labels along paths within a graph to perform the filtering [SWG02].

For similarity search, deciding whether (sub)graphs are isomorph does not suffice. In the case of the two graphs not being identical, similarity search requires an assessment of the degree to which the graphs in question differ from another such that graphs in a database can be sorted by similarity to a query graph.

The comparison of two graphs can be performed by directly considering the structure of the graphs. This approach is, for example, taken by the graph edit distance [BA83] that calculates how many transformations have to be performed to turn one graph into the other, and also by measures that consider common subgraphs or the size of the largest common subgraph [BS98]. While these measures are suitable for small graphs and for graphs with limitations regarding their structure and/or the operations that may be performed (e.g., the degree-2 edit distance for connected, undirected, acyclic trees [ZWS96]), even medium-sized general graphs quickly lead to a query processing time that is bound to overburden the patience of the user.

Akin to content-based image retrieval, feature-based graph similarity models instead derive (approximate) structural information from the graphs and assess the similarity of the graphs based on these features. For example, so-called spectral approaches [Ume88, LWH03] compare graphs based on an eigen-decomposition of the adjacency matrix. The model presented in [PM99] compares two graphs by computing the difference in the number of nodes that have a given connectivity degree. The latter is the basis for the generalized approach described in this short paper. We collected connectivity information along paths in graphs and represent the information in a way that allows graphs to be flexibly compared using the EMD. In a recent related approach, graphs derived from images have been compared using the EMD [GXTL08]. However, the approach uses the EMD to compare the direction of edges/lines that occur in the graph and thus requires the vertices to have a spatial position. The approach described here is devised in a more general way as it does not make such an assumption.

2 Preliminaries

The basic graph-related definitions for concepts used in the rest of the paper are given in this section.

A general graph with at most one edge from one vertex to another is defined via its set of vertices and its edge relation.

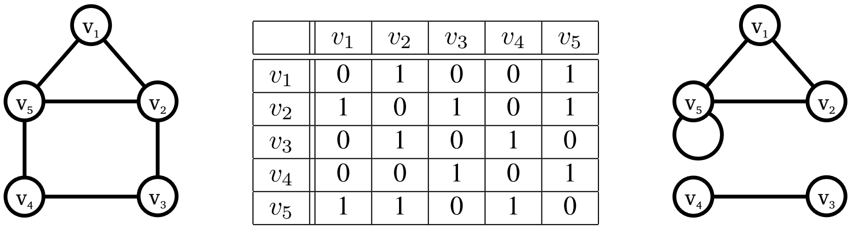
Definition 1 ***(Graph)***
A graph G of size m is a tuple $G = (V, E)$ with vertices $V = \{v_1, \ldots, v_m\}$ and edges $E \subseteq V \times V$.

If a graph does not have single-vertex loops (i.e., the edge relation is irreflexive) and is undirected (i.e., the edge relation is symmetric), it is called simple.

Definition 2 ***(Simple Graph)***
Given a graph $G = (V, E)$, G is simple iff for all $v, w \in V$

$$(v, w) \in E \Leftrightarrow (w, v) \in E \;\; and \;\; (v, w) \in E \Rightarrow v \neq w.$$

All graphs examined in the remainder of this paper are assumed to be simple.



	v_1	v_2	v_3	v_4	v_5
v_1	0	1	0	0	1
v_2	1	0	1	0	1
v_3	0	1	0	1	0
v_4	0	0	1	0	1
v_5	1	1	0	1	0

(a) Graph G_1

(b) Adjacency matrix of G_1

(c) Graph G_2

Figure 1: Example graphs

If all vertices of a graph are connected to all other vertices of the graph by a series of edges, it is called connected.

Definition 3 ***(Connected Graph)***
Given a graph $G = (V, E)$, G is connected iff for all $v, w \in V$:

$$\begin{aligned}(v \neq w) \Rightarrow \quad & (v, w) \in E \\ & \vee \big(\exists v_{i_1}, \ldots, v_{i_{m'}} \in V : \quad (v, v_{i_1}) \in E \wedge (v_{i_{m'}}, w) \in E \\ & \qquad \wedge \forall 1 \leq j \leq m' - 1 : (v_{i_j}, v_{i_{j+1}}) \in E\big)\,.\end{aligned}$$

The graphs in the database are assumed to be connected in this paper. A query graph with missing vertices or edges may however break down into several non-connected components.

The degree of a vertex in a graph is the number of other vertices it is directly connected to.

Definition 4 ***(Vertex Degree Function)***
Given a graph $G = (V, E)$, the outgoing vertex degree function $\delta^G : V \to \mathbb{N}_0$ for G is defined by

$$\delta^G(v) = |\{w \in V | (v, w) \in E\}|.$$

The ingoing vertex degree function can be defined analogously. For the simple graphs of this paper, the two functions are identical and thus do not have to be differentiated here.

The example graph G_1 in Figure 1(a) is a simple, connected graph with 5 vertices and 6 edges. Figure 1(b) gives the adjacency matrix of G_1 where an entry of 1 indicates the existence of an edge while an entry of 0 indicates the absence of an edge between two vertices. As a result of Definition 2, the diagonal entries are all zero and the matrix is symmetric. The degree of a vertex equals the row sum in the adjacency matrix. Vertices v_1, v_3, and v_4 have degree 2 while vertices v_2 and v_5 have degree 3. The graph G_2 is neither simple (due to the loop at v_5) nor connected (due to having two separate components).

3 Graph Similarity Model

In order to find graphs in a database that might be related to a query graph through a process of decay, mutation or generally structural change, a representation of statistical graph features is proposed in Section 3.1 and distance measures suitable for the feature representation are given in Section 3.2. The similarity of two graphs can be assessed by combining these two parts.

3.1 Graph Feature: Degree Co-Occurrence Multisets

A representation of graph features that encodes structural information is required for detecting small structural changes between graphs in a feature-based approach. In this section, statistical features of the vertices that occur in the graphs and their connectivity relationship are discussed. In the simplest form, a graph can be represented by the distribution of the degrees of its vertices as in [PM99]. However, by looking at each vertex separately, one of the core concepts of graphs is ignored. Graphs are useful as they model relationship information between the vertices. Thus, this section proposes to utilize statistical information on the co-occurrence of vertices. In this way, the feature representation encodes which kinds of vertices are connected within a graph – and how frequent this coupling occurs. The co-occurrence concept can be generalized by looking at occurrences along paths in the graph and noting which kinds of vertices occur close to each other / in sequence. In the following definitions, the generalized co-occurrence concept is formally introduced on the basis of vertex degrees as this information is common to all graphs. If other categorical information (e.g., vertex class labels) is available, the approach could be adapted to incorporate that information.

Definition 5 ***(Simple Vertex Path)***
With $G = (V, E)$ as a graph, the $(m + 1)$-tuple $(v_{i_0}, \ldots, v_{i_m}) \in V^{m+1}$ is a simple (non-looping) vertex path of length m in G iff

$$\forall 0 \leq j < j' \leq m : v_{i_j} \neq v_{i_{j'}}$$

and

$$\forall 0 \leq j < m : (v_{i_j}, v_{i_{j+1}}) \in E.$$

The set of all simple paths of length m in G is denoted as P_m^G.

For the cases of path lengths $m = 0$ and $m = 1$, sets P_0^G and P_1^G equal the set of vertices V and the set of edges E. Using the set of simple paths of length m, a co-occurrence multiset of degree m captures the frequencies of vertex class (here, vertex degree) sequences.

Definition 6 ***(Vertex Degree Co-occurrence Multisets)***
With $G = (V, E)$ as a graph, the Vertex Degree Co-Occurrence Multiset D_m^G of degree m for graph G is defined as a tuple

$$D_m^G = \left(DS_m^G, f_m^G\right)$$

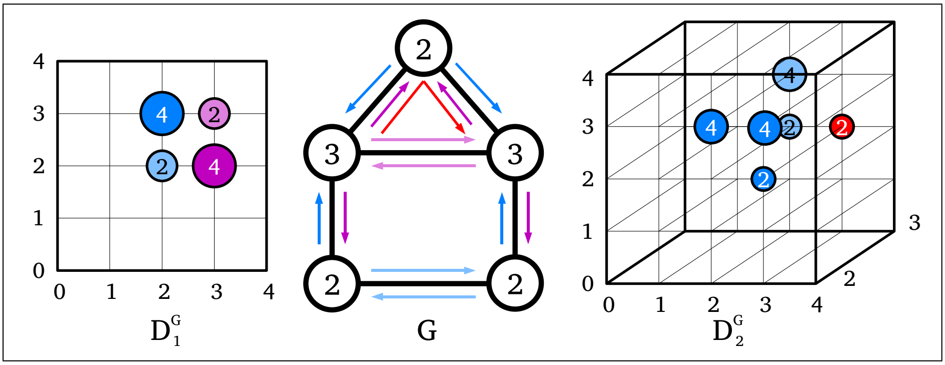


Figure 2: Visualization for the multiset feature representation of a graph G

where

$$DS_m^G = \{(\delta^G(v_{i_0}), \dots, \delta^G(v_{i_m})) \mid (v_{i_0}, \dots, v_{i_m}) \in P_m^G\}$$

is the set of all vertex degree sequences occurring on paths of length m in G and

$$f_m^G(dg_0, \dots, dg_m) = \left|\{(v_{i_0}, \dots, v_{i_m}) \in P_m^G \mid \forall 0 \leq j \leq m : dg_j = \delta^G(v_{i_j})\}\right|$$

the frequency function of such sequences in G.

The set DS_m^G abstracts from individual vertices by only considering their type (i.e., vertex degree in this example). The degree m of the multiset is not related to the degree of the vertices in the graphs but only to the length of the examined paths.

As an example, the graph G_1 in Figure 1(a) has five paths of length $m = 0$ (i.e., $P_0^{G_1} = V = \{v_1, \dots, v_5\}$). The occurring vertex degrees are $DS_0^{G_1} = \{(2), (3)\}$ with frequencies $f_0^{G_1}(2) = 3$ and $f_0^{G_1}(3) = 2$. For $m = 1$, there are twelve paths (i.e., two per edge). The combinations of vertex degrees occurring along those paths are $DS_1^{G_1} = \{$(2, 2), (2, 3), (3, 2), (3, 3)$\}$. The frequencies of those paths are $f_1^{G_1}(2,2) = 2$, $f_1^{G_1}(2,3) = 4$, $f_1^{G_1}(3,2) = 4$, and $f_1^{G_1}(3,3) = 2$. The set of vertex degree sequences $DS_2^{G_1}$ is of cardinality 6 and $DS_3^{G_1}$ of cardinality 8. The experiments in Section 4 show good results for m as low as 2. For greater lengths, techniques such as random path sampling could be applied to speed up the feature extraction process.

Figure 2 shows a visualization of the co-occurrence multisets D_1^G (on the far left) and D_2^G (on the far right) in the form of bubble charts. The x, y, and z axes denote the degree of the first, second, and third vertex on a path in G. The size of the bubbles is proportional to the frequency of the according vertex degree sequences that is also denoted inside the bubble. For $m = 1$, the short arrows next to the graph in the middle of the figure show all paths (i.e., edges) that contribute to the multiset D_1^G. The long arrow in the upper section of the graph shows a path that contributes to the bubble at coordinate 3-2-3 in the far right of the figure representing D_2^G.

3.2 Similarity Measure

With the above definitions, a co-occurrence multiset can be associated with each graph in the database and with the query graph. Graph similarity can then be assessed in terms of co-occurrence multisets that capture statistical information on the structure of the graphs. Next, we describe how this feature representation can be compared via distance measures.

3.2.1 Element-Wise Multiset Comparison

A first approach is to treat the multisets as sparse representations of high-dimensional vectors. Since the multisets are finite, norm-based distance measures such as the L_p distances can be adapted to compare two graphs represented by such multisets.

Definition 7 ***(L_p Distance on Vertex Degree Co-Occurrence Multisets)***
Given two graphs G_1 and G_2 with associated vertex degree co-occurrence multisets $D_m^{G_1} = (DS_m^{G_1}, f_m^{G_1})$ and $D_m^{G_2} = (DS_m^{G_2}, f_m^{G_2})$ according to Definition 6, the L_p distance between the two multisets is defined as

$$d_{L_p}(D_m^{G_1}, D_m^{G_2}) = \left(\sum_{ds \in (DS_m^{G_1} \cap DS_m^{G_2})} |f_m^{G_1}(ds) - f_m^{G_2}(ds)|^p \right.$$
$$+ \sum_{ds \in (DS_m^{G_1} - DS_m^{G_2})} |f_m^{G_1}(ds)|^p$$
$$\left. + \sum_{ds \in (DS_m^{G_2} - DS_m^{G_1})} |f_m^{G_2}(ds)|^p \right)^{1/p} .$$

In the case of $m = 0$ and $p = 1$, the similarity model reflects the one of [PM99] where graphs are compared using simple vertex degree histograms and the Manhattan distance.

3.2.2 Transformation-Based Multiset Comparison

Another possibility is to employ similarity measures that can inherently cope with weighted feature sets instead of just feature vectors such as the EMD [RTG98]. For this purpose, we first introduce the feature signatures used as an input for the EMD, followed by the definition of the EMD.

Definition 8 ***(Feature Signatures)***
Given an object o represented by features $f_1, ..., f_k$ in a feature space FS, and an n-clustering $C_1, ..., C_n$ of these features, the feature signature s^o of the object o is defined as a finite set of tuples from $FS \times \mathbb{R}$:

$$s^o = \{(r_1^o, w_1^o), ..., (r_n^o, w_n^o)\}$$

where $r_i^o \in FS$ represents the feature cluster C_i and $w_i^o = \frac{|C_i|}{k}$ is the relative cardinality or weight/mass of the according cluster.

The EMD itself is defined as a linear optimization problem. The similarity between two signatures s^o and s^q is defined as the minimal cost for transforming the signature s^o into the signature s^q where a ground distance gd determines the cost of transforming/moving a unit of mass from a cluster of the first signature to a cluster of the second signature. Linear constraints on the movement of mass describe the set of feasible combinations of transformations.

Definition 9 ***(Earth Mover's Distance (EMD)***
Given two signatures s^o, s^q, and a ground distance gd, the EMD between s^o and s^q is defined as the minimum over feasible transformations $F \in \mathbb{R}^{|s^o| \times |s^q|}$:

$$EMD_{gd}(s^o, s^q) = \min_F \left\{ \frac{1}{\tilde{w}} \sum_i \sum_j F[i,j] \cdot gd(r_i^q, r_i^o) \right\}$$

under linear constraints

$$\forall i,j : F[i,j] \geq 0$$

$$\forall i : \sum_j F[i,j] \leq w_i^q$$

$$\forall j : \sum_i F[i,j] \leq w_j^o$$

$$\sum_i \sum_j F[i,j] = \tilde{w}$$

with $\tilde{w} = \min\{\sum_{i=1}^n w_i^o, \sum_{i=1}^m w_i^q\}$.

Intuitively, the first group of constraints ensures that earth is only moved from clusters of s^o to clusters of s^q, the second and third group of constraints ensures that no more mass is removed from or moved to the clusters than their respective weight permits and the last constraint ensures that in total as much mass as possible is moved.

The similarity of two graphs can be assessed using the EMD by defining a transformation from the multisets to the signatures of the EMD. The co-occurrence multisets are a close match to the signatures that the EMD takes as its input.

Definition 10 ***(Feature Signatures of Graphs)***
Given a graph G with an associated vertex degree co-occurrence multiset $D_m^G = (DS_m^G, f_m^G)$, the feature signature s_m^G of G for comparison with the Earth Mover's Distance is defined as

$$s_m^G = \left\{ (r,\ w) \mid r \in DS_m^G \wedge w = \frac{f_m^G(r)}{|DS_m^G|} \right\}.$$

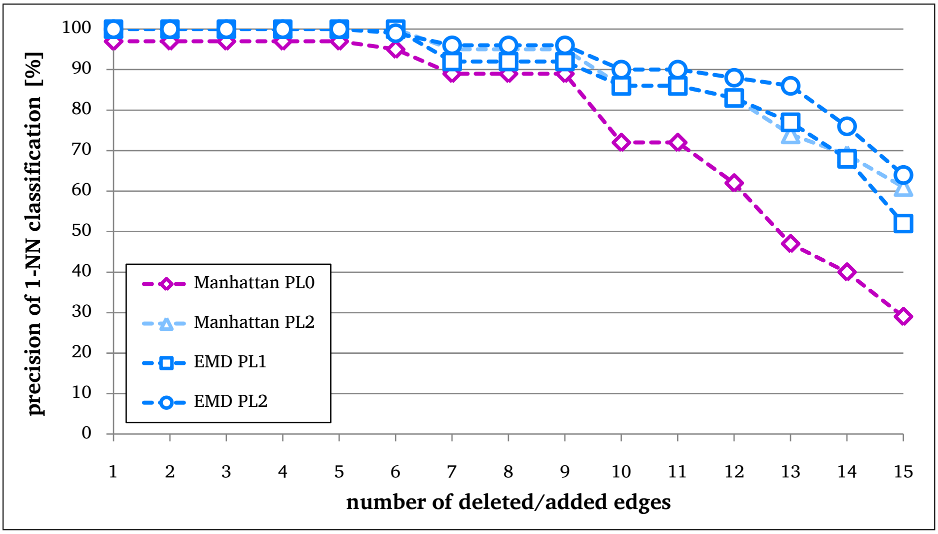


Figure 3: Adding/deleting edges at random; $|DB| = 1000$

The cost for transforming one degree sequence into another one can be defined via a ground distance function. In the simplest case, the sequences can be treated as vectors from ${\mathbb{N}_0}^m$ and compared using Minkowsi distance measures d_{L_p}. In this way, a degree sequence that deviates from another for example by starting with a degree of 3 instead of 4 will induce a lower transformation cost than one that starts with 1 instead of 4. Distance measures such as the Edit Distance, which take the sequential character of the representatives r into account, could also be employed. For undirected graphs, the fact that each sequence of vertex degrees appears twice in both directions should be accounted for by adjusting either the signature definition or the ground distance.

4 Preliminary Experimental Results

For the preliminary experiments shown here, a number of synthetic graph databases of differing cardinalities were created using the method detailed in [VL05] based on sequences of vertex degrees following a power-law distribution with modifications to ensure that the graphs are connected and simple. All graphs randomly generated in this fashion had 100 vertices and 150 edges. The average vertex degree was set to 3, resulting in power-law graphs with a relatively large number of low degree vertices and a relatively low number of high degree vertices.

In the first set of experiments, 100 graphs were randomly chosen from the database as the basis for 15 query graphs each that represent different levels of structural deviation regarding the edge relation. For each of the 15 levels, a random edge was either inserted or deleted with equal probability. Not accounting for edges that may have been deleted

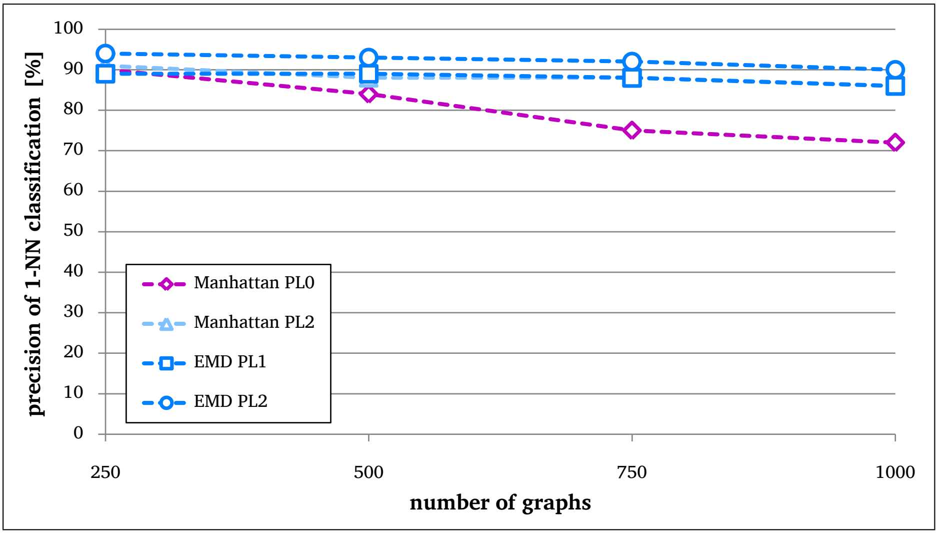


Figure 4: Adding/deleting edges at random; 10 edges added/deleted

and consecutively been added again, up to 10% of the the edge relation may have been changed in this process.

The vertical axis of Figure 4 shows how often the graph on which the query graph was based was identified as the most similar one out of all 1000 graphs in the database. A greater path length (denoted as PL in the figure) for the vertex degree co-occurrence multisets results in a similarity model that is more robust with regard to the structural change for this experiment. The greater the structural difference, the more can the multisets based on longer paths distinguish themselves from those of lower degree. The Manhattan distance on simple vertex degree histograms (cf. [PM99]) is always outperformed by the multisets of higher degree (i.e., based on longer paths) in this experiment. The EMD with a Manhattan ground distance slightly outperforms the Manhattan distance for equal path lengths. The EMD for path length zero is not plotted here, as the results equal those of the Manhattan distance in the case of a one-dimensional feature space and Manhattan ground distance.

Figure 4 shows that the higher degree multisets are also less influenced by the cardinality of the database. Even though the database size on the right is four times the size of the database on the left, the number of times that the original graph from the database is not identified as the most similar one to the query graph only slightly increases from 9 out of 100 to 14 out of 100 for the EMD with path length two. The degree histogram approach jumps from 10 out of 100 to 28 out of 100 for the same increase in database size.

The two figures 4 and 4 show the results of according experiments when considering structural change that is not limited to the edge relation. Instead, random vertices were removed together with their adjacent edges. As is to be expected due to the greater level of structural change, all approaches show a faster decrease of the precision with which they can

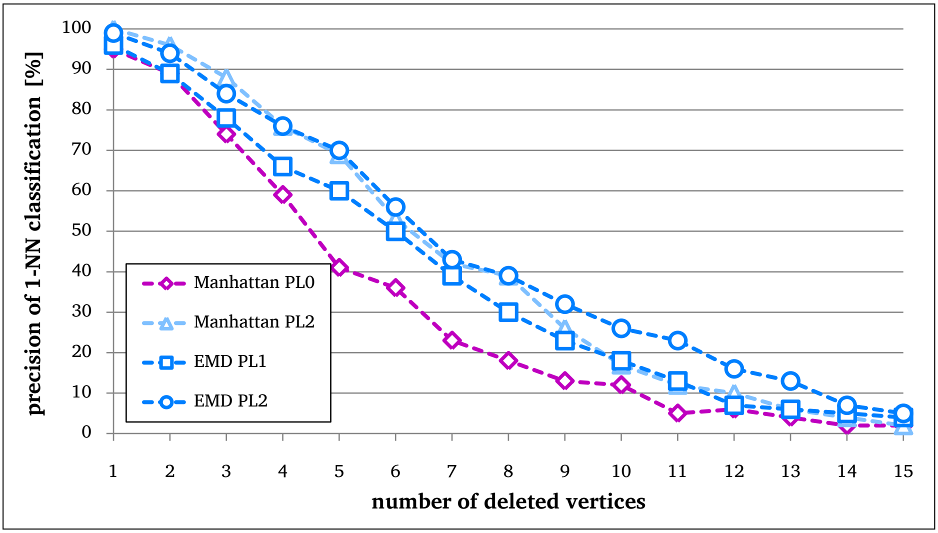


Figure 5: Deleting vertices at random; $|DB| = 1000$

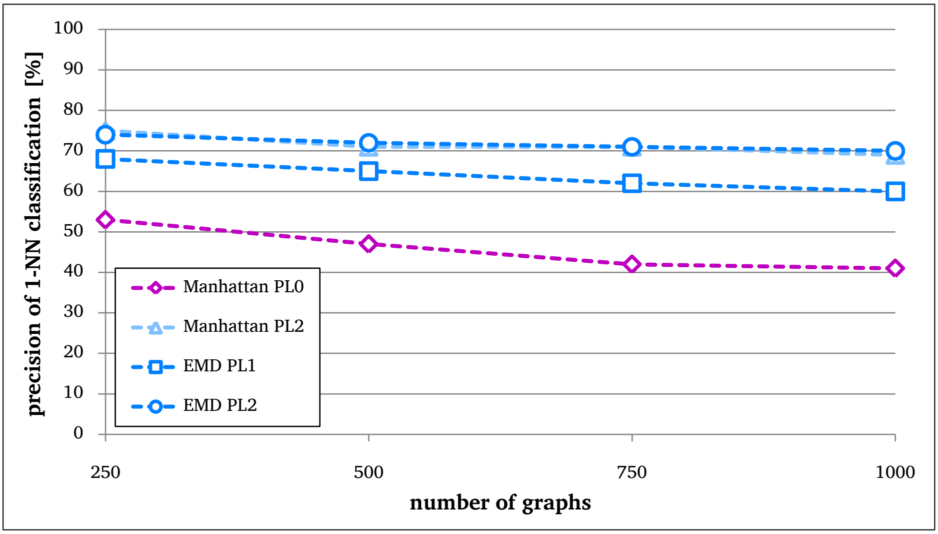


Figure 6: Deleting vertices at random; 5 deleted vertices

identify the original graph in the database. Greater path lengths still produced better results in these experiments while the EMD with a Manhattan ground distance was only able to outperform the normal Manhattan distance for more severe levels of structural change in this case.

5 Conclusion and Outlook

In this short paper, we showed how complex data objects in the form of graphs can be compared using the EMD by defining a suitable representation of graph features that capture statistical information regarding the structure of the graphs. In this way, it is possible to identify graphs that resulted from some other graph through a process of structural change without having to resort to typically very expensive similarity measures that directly take the graph structure into account.

The general viability of the approach was shown using a Manhattan ground distance for the EMD together with vertex degrees as the sole information regarding the vertices. For this ground distance a projection-based lower bound for the EMD [CG97] can be applied in a filter step in order to gain efficiency, especially for higher degrees of the multiset. Also the EMD-L1 algorithm from [LO07] can be employed to speed up retrieval. While the preliminary results using this simple ground distance were generally good, the Manhattan ground distance potentially limits the benefits of longer co-occurrence sequences that are used as signature component representatives for the EMD. Other ground distances that take the sequence character of the feature representatives (i.e., sequences of vertex degrees in this case) into account may present an opportunity to further improve the technique.

6 Acknowledgments

This work is partially funded by DFG grant SE 1039/1-3.

References

[BA83] Horst Bunke and Gudrun Allermann. Inexact Graph Matching for Structural Pattern Recognition. *Pattern Recognition Letters*, 1(4):245–253, 1983.

[BS98] Horst Bunke and Kim Shearer. A Graph Distance Metric Based on the Maximal Common Subgraph. *Pattern Recognition Letters*, 19(3-4):255–259, 1998.

[CG97] Scott D. Cohen and Leonidas J. Guibas. The Earth Mover's Distance: Lower Bounds and Invariance under Translation. Technical Report STAN-CS-TR-97-1597, Stanford University, 1997.

[GXTL08] Xinbo Gao, Bing Xiao, Dacheng Tao, and Xuelong Li. Image Categorization: Graph Edit Distance + Edge Direction Histogram. *Pattern Recognition*, 41(10):3179–3191, 2008.

[HHEW04] Lei He, Chia Y. Han, Brian Everding, and William G. Wee. Graph Matching for Object Recognition and Recovery. *Pattern Recognition*, 37:1557–1560, 2004.

[LO07] Haibin Ling and Kazunori Okada. An Efficient Earth Mover's Distance Algorithm for Robust Histogram Comparison. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 29(5):840–853, 2007.

[LWH03] Bin Luo, Richard C. Wilson, and Edwin R. Hancock. Spectral Embedding of Graphs. *Pattern Recognition*, 36(10):2213–2230, 2003.

[PM99] Apostolos N. Papadopoulos and Yannis Manolopoulos. Structure-Based Similarity Search with Graph Histograms. In *Proceedings of the International Workshop on Database and Expert Systems Applications (DEXA)*, pages 174–178, 1999.

[RTG98] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. A Metric for Distributions with Applications to Image Databases. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 59–66, 1998.

[SKK04] Thomas B. Sebastian, Philip N. Klein, and Benjamin B. Kimia. Recognition of Shapes by Editing Shock Graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 26(5):550–571, 2004.

[SWG02] Dennis Shasha, Jason T. L. Wang, and Rosalba Giugno. Algorithmics and Applications of Tree and Graph Searching. In *Proceedings of the ACM Symposium on Principles of Database Systems (PODS)*, pages 39–52, 2002.

[Tak04] Yoshimasa Takahashi. Chemical Data Mining Based on Non-terminal Vertex Graph. pages 4583–4587, 2004.

[Ume88] Shinji Umeyama. An Eigendecomposition Approach to Weighted Graph Matching Problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(5):695–703, 1988.

[VL05] Fabien Viger and Matthieu Latapy. Efficient and Simple Generation of Random Simple Connected Graphs with Prescribed Degree Sequence. In *Proceedings of the International Computing and Combinatorics Conference (COCOON)*, pages 440–449, 2005.

[ZWS96] Kaizhong Zhang, Jason T.-L. Wang, and Dennis Shasha. On the Editing Distance Between Undirected Acyclic Graphs. *International Journal of Foundations of Computer Science*, 7(1):43–58, 1996.