ISSN: 2834-7706

Research Article

Journal of Mathematical Techniques and Computational Mathematics

Assessing the Quality in the Detection of Similar Complex Data Structures in Large-Scale Datasets

Pierpaolo Massoli*

Directorate for Methodology and Statistical Process Design (DCME), Italian National Institute of Statistics (ISTAT), Italy

***Corresponding Author**
Pierpaolo Massoli, *Directorate for Methodology and Statistical Process Design (DCME), Italian National Institute of Statistics (ISTAT), Italy*

Submitted: 2024, Jul 05; **Accepted:** 2024, Jul 24; **Published:** 2024, Aug 08

Citation: Massoli, P. (2024). Assessing the Quality in the Detection of Similar Complex Data Structures in Large-Scale Datasets. *J Math Techniques Comput Math, 3*(8), 01-09.

Abstract

The existence of complex data structures in today's data collections requires appropriate approaches driving the scientific community towards elaborating more efficient methods for data analysis. Graph Theory can be effectively applied for mathematical modeling these structures as is the case in network analysis. The search for similar networks may therefore be viewed as a graph matching problem, which poses a fundamental challenge in real-world applications. This study investigates the quality of the detection of similar complex data structures which follows a novel approach introduced recently. The detection approach employs some basic concepts from the Graph Theory for leveraging the Locality Sensitive Hashing to efficiently address the graph matching problem for finding isomorphic graphs as well as the common subgraph embedded within them. This method may generate false duplicates which affect the accuracy of the solution so that even the finest tuning of the hyperparameters does not guarantee high levels of accuracy. This study therefore proposes an in-depth investigation of crucial aspects of the detection approach in order to assess the accuracy of the same. The similarity of the detected pairs of similar graphs is analyzed as well as the critical aspect of the hashing step is investigated by bootstrapping the solution in order to assess its statistical properties. A real-world case study is considered to validate the potential of the proposed approach.

Keywords: Network Modelling, Graph Matching, Locality Sensitive Hashing, Bootstrap

1. Introduction

The vast amount of today's data available for scientific research requires the development of increasingly efficient approaches for data analysis. A task of great interest for practical applications is identifying similar complex data structures in large-scale datasets. A widespread approach in the literature used to accomplish this task is the Graph Matching (GM) problem, which searches for an alignment between the vertex sets of graphs by preserving the common structure within them. This is posed as minimizing edge disagreements over all possible vertex alignments. Graph matching has various applications in diverse fields, such as pattern recognition machine learning bioinformatics neuroscience social network analysis and knowledge discovery in natural language processing [1-10]. In all these cases, the problem of finding an alignment between networks can be thought of as a variant of the GM problem by selecting the appropriate objective function to be optimized. The well-known graph isomorphism problem is a special case of the GM problem, which aims to find a bijection between the vertices of two graphs that exactly pre-serves the edge structure. The GM is generally equivalent to the NP-hard quadratic assignment problem, which is a challenging problem even though polynomial- time algorithms are applicable in the case of nearly isomorphic graphs [11,12]. Although an extensive review of the literature pertaining to the GM problem focuses on the pattern recognition topic, it is rather straightforward to accept that graph matching can also be addressed as a similarity search problem, with nearest neighbors graphs detected according to a predefined metric [13-15]. Due to the fact that in large-scale datasets, pairwise comparisons of the input data can hinder the majority of state-of-the-art methods, the use of approximate nearest neighbors search method is more efficient [16]. The idea behind this study is to leverage the Locality Sensitive Hashing technique to detect similar objects in high-dimensional spaces by tolerating the presence of false duplicates [17-21]. In real-world applications, the concept of a network, which is used to describe a complex system of entities, is more popular as it is better understood even by the non-scientific community. A network is a set of objects called nodes or vertices that are connected to one another by edges or links. In mathematics, networks are often referred to as graphs, so the theoretical background of Graph Theory can be used for network modeling as well. One of the most important issues in network analysis is detecting similar structures embedded in networks, similar to determining similar subgraphs in a collection of graphs. In real-world networks, nodes may have attributes that are useful for network structure exploration [22]. Exploring large-

scale datasets containing networks of different dimensions is a challenging task, which is often faced in practical applications. In these cases the leveraging of Locality Sensitive Hashing approach is a valid solution with respect to the computational effort [23]. This article adopts this approach for the detection of isomorphic networks as well as sub-networks embedded intp larger graphs in accordance with a suitable metric for graph matching problems. Although this method is computationally efficient and accurate a quality assessment of the same is mandatory. Relevant properties pertaining to the solution found as well as the hashing algorithm which is the key concept of the proposed approach are investigated in the this study.

2. Theoretical Background

In order to introduce the fundamental aspects of the detection process to the reader, some notions from the Graph Theory as well as the basic concepts of the Locality Sensitive Hashing technique are reported in this section.

2.1 Graph Theory Background

A graph $G = (V, E)$ with $i = 1, 2, \ldots, n$ vertices $v_i \in V$ and $j = 1, 2, \ldots, m$ edges $e_j \in E \subset V \times V$ is *undirected* If the edges have no direction and simply connect pairs of vertices. The graph is said to be *connected* if every pair of vertices in the graph is connected, i.e. there is a path between every pair of vertices. The graph is said to be *complete* or fully *connected* if each vertex is connected to all other vertices so that the set E is constituted by $m = n(n-1)/2$ edges as is the case in undirected graphs. The geometric structure of the graph is summarized by its *adjacency* matrix $\mathbf{A} = \{a_{kh}\}$ defined as follows:

$$a_{kh} = \begin{cases} 1 & \text{if } v_k \text{ is adjacent to } v_h \\ 0 & \text{if } v_k \text{ is not adjacent to } v_h \text{ or } v_k \equiv v_h \end{cases} \quad (1)$$

This matrix is symmetric if the graph is undirected. An alternative formulation often adopted in particular for large graphs is the *adjacency list* in which each pair of vertices connected by an edge of the graph is listed by row in a table. The graph is said to be *weighted* if there exists a real number w_{kh} (weight) related to each edge e_{kh} in that the adjacency matrix $\mathbf{W} = \{w_{kh}\}$ is as follows:

$$w_{kh} = \begin{cases} w_{kh} & \text{if } v_k \text{ is adjacent to } v_h \\ 0 & \text{if } v_k \text{ is not adjacent to } v_h \text{ or } v_k \equiv v_h \end{cases} \quad (2)$$

A *simple closed path* of length l starting from vertex i and returning to the same is a sequence of distinct vertices connected by l edges. In a weighted graph the simple closed path of minimum cost is the sequence of edges related to the smallest value of the sum of their weights. A *complete subgraph* or *clique S(G)* is a group of fully connected vertices belonging to the vertices set of the graph. The *Depth-First* Search (DFS) algorithm is an algorithm to explore a graph. The DFS is very appropriate for identifying the connected components into a graph. If the graph has disconnected components, DFS can be used to explore and locate each connected component as well as complete subgraphs effectively.

2.2 Graph Matching Basics

The problem of the graph matching between the graphs $G_i = (V_i, E_i)$ and $G_j = (V_j, E_j)$ is generally formulated as follows:

$$\underset{\mathbf{P} \in \mathbf{\Pi}}{\operatorname{argmin}} \quad ||\mathbf{A}_i - \mathbf{P}\mathbf{A}_j\mathbf{P}^T||_F \quad (3)$$

where $\mathbf{A}_i$ and $\mathbf{A}_j$ are the adjacency matrices of the graphs to compare. The objective is to find the matrix $\mathbf{P}$ which represents the optimal assignment. The general formulation of this problem is NP-hard even though in some real-world applications turns into a linear problem which is solvable in $O(n^3)$ for an assignment of n vertices.

2.3 Locality Sensitive Hashing Fundamentals

In data science Locality Sensitive Hashing (LSH) refers to a method designed for an approximate similarity search in high-dimensional spaces where traditional search methods become computationally expensive. There are several metrics that LSH encompasses for finding near-duplicates by means of a suitable family of hash functions $h(\bullet)$ which establish a relation between two input data points $(\mathbf{x}_k, \mathbf{x}_h) \in \mathbf{X}$ and the probability of sharing the same hash code: $sim(\mathbf{x}_k, \mathbf{x}_h) = P[h(\mathbf{x}_k) = h(\mathbf{x}_h)]$. The choice of the hash function determines the metric to approximate. Every family associates input data to integers which are thought of as being *buckets* with the purpose of hashing is to group similar data points together into the same bucket so that neighboring data fall into the same bucket with a high probability while data which are likely to be distant in the input space belong to different buckets. In a database context, this facilitates the detection of pairwise similar observations in accordance with varying degrees of similarity. In this study the LSH-family known as *minhash* tailored for evaluating the similarity between sets by approximating the *Jaccard index* is adopted. In order to use this specific LSH-family, each input object is transformed into a set of features called *shingles*. As an example, if the data objects in the input dataset were texts they would be broken down into k-shingles which are sequences of k consecutive characters so that each text would be transformed into a set of shingles. As is the case every input data has to be transformed into a set of appropriate features which will be referred to as shingles. Every shingle $\mathbf{s}$ is subsequently hashed into an integer number by using a hash function $h(\mathbf{s})$. By applying this function to every shingle belonging to the set in which the input object has been converted it becomes a set of integer numbers. The minimum value of these integers is the *minhash* code pertaining to the input object. By means of a sequence of H randomly generated hash functions $h_i(\mathbf{s})$, the input dataset is transformed into a dataset of *signatures* which are sequences of H i.i.d. hash codes. As a result the input dataset containing N objects of varying dimension is transformed into a $(N \times H)$ *signature matrix* which is elaborated in the section which follows.

2.4 Near-Duplicates Search

Subsequent to the generation of the aforementioned matrix each signature is shrunk into B bands in order to speed up the search for near-duplicates. Each band consists of R adjacent combined hash codes so that the relation $H = BR$ holds. Similar input objects are finally detected by sorting the $(N \times B)$ *banded*

signature matrix and sequentially scanning it B times. Every pair of consecutive signatures with at least one corresponding equal band indicates a pair of near-duplicate input objects. The probability of there being a pair of similar objects with a similarity value σ is given by:

$$\pi = 1 - (1 - \sigma^R)^B \qquad (4)$$

It is widely reported in the literature that the LSH is an approximate method which may give rise to *false duplicates* in the solution. The rate of the same as up to now being controlled solely by means of an appropriate tuning process of the hyperparameters.

2.5 Bootstrap Method

The bootstrap method is a powerful statistical technique used for estimating the distribution of a sample statistic by resampling with replacement from the original dataset [24,25]. This method is particularly valuable when the theoretical distribution of a statistic is complex or unknown. By repeatedly sampling from the data and recalculating the statistic for each sample, the bootstrap method allows for the approximation of the sampling distribution, which can be used to construct confidence intervals perform hypothesis testing, and assess the uncertainty of the estimates. Key advantages of the bootstrap method include its flexibility and minimal assumptions. It can be applied to a wide range of statistical problems, including those involving complex models and small sample sizes. Additionally, it does not require assumptions about the underlying distribution of the data, rendering it a robust tool in both parametric and non-parametric contexts. The bootstrap method is a versatile and widely-used approach in modern statistics, enabling more accurate and reliable inference when traditional methods are infeasible or insufficient.

3. Detection of Similar Network Structures

The proposed algorithm is devised for detecting isomorphic networks as well as similar sub-networks embedded in different ones contained in a large dataset. The main steps of the algorithm are described in this section.

3.1 Input Networks

The input dataset contains N networks which are mathematically described as being fully connected undirected weighted graphs of n vertices. The number of vertices is variable so that there are networks of different dimensions in the dataset. Every vertex is related to a sequence of K categorical variables (nodal attributes) $a^{(1)}\{a_1^{(1)}, a_2^{(1)}, \ldots, a_K^{(1)}\}$ called the *profile* of the vertex v_i (i = 1, 2, . . . , n). A sketch of input Network is reported in Figure 1. The edges e_{ij} of the graph are related to real numbers $w_{ij} \in [0,1]$ which indicate the algebraic complement to the relative frequency of the pair of the observed profiles of the node i and the node j with respect to the total number of profiles in the entire dataset. The similarities of interest are calculated by using the attributes related to the graphs. The Jaccard similarity is calculated by appropriately considering the triangles of the graphs to be compared. As a consequence a pair of networks which share the same profiles corresponds to a value of the Jaccard similarity equal to 1 while this value decreases as the number of profiles in common decreases.

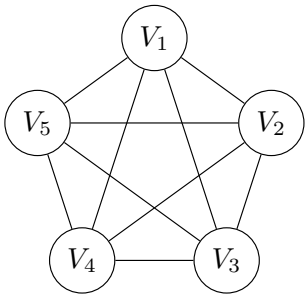

Figure 1: Input network of n = 5 vertices

For an isomorphism between two graphs, there has to be a one-to-one correspondence between their vertices while preserving the links between them at the same time. Only the networks having the same number of nodes as well as the same profiles are considered isomorphic. The special case of two networks having the same node profiles but a different number of nodes is emphasized by the proposed approach. Therefore, the cases which remain reveal the correspondence between subgraphs.

3.2 Network Hashing

Every possible profile j = 1, 1, . . . , P is coded by randomly coupling it with a *unique* integer number $x_j \sim U[0, m-1]$ of fixed length L in bits so that the total number of possible integers is equal to $m = 2^L$. This length depends on the number of all possible profiles $P = \prod_{h=1}^{K} |a_h|$ where $|\cdot|$ is the cardinality of the categorical variable. For each graph in the input dataset, the list of all the shingles of length 3, i.e. *triangles* of minimum cost is created so that there is a resulting list of n triangles pertaining to a graph of n nodes. Every triangle $\mathbf{t}_i$ is constituted by a triplet of integers $\{xi, x_h, x_k\}$ where i, h, k = 1, 2, . . . , n ($i \neq h \neq k$) which is hashed on the basis of the following:

$$h_q(\mathbf{t}_i) = \sum_{k=1}^{3} [(\gamma_q + \mathbf{t}_i(k)\, \alpha_q^{(k-1)}) \mod m] \qquad (5)$$

where $\mathbf{t}_i(1) = x_i$, $\mathbf{t}_i(2) = xh$ and $\mathbf{t}_i(3) = x_k$. The parameters (α_q, γ_q) have to satisfy the statistical requirements of randomness and uniformity as well as the *universal hashing* requirement in order to reduce the number of collisions as much as possible [26]. It is straightforward that in order to map all the triangles into m different hash codes the number of possible profiles is limited by the condition $P \leq \sqrt[3]{m}$ implying a relation between profiles and the minimum memory bit-space required 1. The function reported in Equation 5 is applied to all the i = 1, 2, . . . , n triangles in the list $\mathbf{T}_{Gj}$ related to the graph G_j in the input dataset. The minimum value of the integers in the list is the *minhash* code of the network. By generating q = 1, 2, . . . , H i.i.d. hash functions every graph is identified by a sequence of H minhash codes (signature). Subsequent to the transformation of the input dataset of N graphs into a ($N \times H$) signature matrix the search for near-duplicate graphs is carried out as described in Section 2.

The number of bits necessary is equal to: $L \geq \lceil 3 \log_2 P \rceil$ (highest nearest integer).

3.3 Optimization of the Solution

The LSH-family of minhash approximates the pairwise Jaccard similarity between the graphs by considering all the triangles they have in common. The solution set should be composed solely by all the pairs with a high probability of being similar with a high degree of similarity. Due to the probabilistic nature of the LSH, the presence of false duplicates must be controlled by carefully setting the parameters $\{H,B,R\}$. Their setting is generally a critical aspect of the nearest neighbors search in that an inappropriate setting could compromise the solution. The parameters in the algorithm proposed here are therefore set in order to achieve an almost zero false negatives rate in opposition to a probable higher false positives rate. In order to lower the rate of false positives, the number of the pairs detected can be reduced by evaluating the Jaccard index of every detected pair directly and therefore by filtering out all the pairs whose similarity satisfies a desired criterion. By setting $|G_i|$ and $|G_j|$ the number of unique traingles in the graphs *i* and *j* respectively, the condition

$$J(G_i, G_j) = \frac{min(|G_i|, |G_j|)}{max(|G_i|, |G_j|)} \tag{6}$$

reduces the solution to pairs having a similarity eqeual to 1 as well as pairs with a subgraph entirely embedded in the larger graph of the pair.

3.4 Evaluating the Solution

The solution set is split into the partitions which follow:

- $\mathbf{S_1}$: is the subset of pairs of *isomorphic* graphs G_i and G_j $(i \neq j)$ with the Jaccard index $J(G_i, G_j) = 1$ and $|V_i| = |V_j|$;
- $\mathbf{S_2}$: is the subset of pairs G_i and G_j $(i \neq j)$ with the Jaccard index $J(G_i, G_j) = 1$ and $|V_i| \neq |V_j|$. The graphs in every pair of this set share the same node profiles;
- $\mathbf{S_3}$: is the subset of pairs G_i and G_j $(i \neq j)$ with the Jaccard index $J(G_i, G_j) < 1$. The graphs in every pair of this set have a matching subgraph;

The estimation of the probability of there being a pair with a given degree of similarity as described in Equation 4 does not imply a reliable setting of the LSH hyperarameters. Hashing collisions are inevitable even when the modulus m is large enough with respect to the number of profiles P which have to be mapped as indicated in Section 3.

Collisions may give rise to false duplicate networks, therefore it is worth investigating to what extent the hashing method in Equation 5 may affect the accuracy of the proposed approach. In this study the three aforementioned sub- sets which compose the solution are analyzed by performing different tests. The first testing approach provides the comparison of the two distributions pertaining to the average of the edge-weigths of the graphs. The asymptotic two-sample Kolmogorov- Smirnov test is carried out separately on every subset. The objective being that of evaluating whether the distribution on the left pertaining to every first graph in the

This condition is the same as requiring the maximum value pf the overlap between the teo graphs of the pair defined as folloes: $overlap = |\boldsymbol{G}_i \cap \boldsymbol{G}_j|/max(|\boldsymbol{G}_i|, |\boldsymbol{G}_j|)$

pairs is the same as that on the right. The second investigation provides the evaluation of the confidence interval (CI) for the average number of collisions caused by the hashing algorithm of Equation 5. The average number of collisions is estimated by generating a signature of *H* hashes for every distinct triangle belonging to the graphs in each subset and by checking whether two different triangles share the same hash code. By repeating this test *H* times, the overall average value is considered as being the parameter of interest. Subsequent to the determination of these parameters their confidence intervals are estimated by using the *Bootstrap* method for each subset of the solution. Similarly to this last test, the confidence intervals of: (1) average number of equal bands between the signatories, (2) average number of triangles in common for each detected pair of graphs as well as (3) average number of profiles in common are estimated via bootstrap method for each subset.

4. Application to a Statistical Population Register

The detection of complex data structures contained in statistical registers is an inter- esting case study for testing the potential of the proposed approach. The data source is a collection of socio-economic individual attributes describing the living conditions of a population referred to a specific time period obtained by integrating several sta- tistical registers and administrative data pertaining to: demographic characteristics, occupation, education and income.

4.1 The Input Dataset

Input data comprises a subset of the entire available aforementioned dataset of a spe- cific territory. A population of 940535 people is grouped into N = 253286 households by means of an identification number. In this case the number of households was restricted to groups of n 3 members only, so that the complex data structures to investigate concern different number of people ranging from n = 3 to n = 14 members. The list of the attributes of each individual is reported in Table 1. As a consequence

N	Variable	Description	Number of classes
1	GENDER	Gender of the household member	2
2	AGE	Age of the household member (in classes)	4
3	CITIZEN	Citizenship of the household member	2
4	EDUCATION	Level of education of the household member	4
5	MAIN SOURCE	Main source of income of the household member	7

Table 1: Attributes in the Input Dataset

the total number of possible profiles is equal to $P = 448$.

4.2 Complex Data Structures Hashing

Representing households as networks or graphs makes sense in the context about to be described. Every household is a fully connected undirected graph. Nodal profiles are the observed combinations of the attributes reported in Table 1. In accordance with the number of profiles P the appropriate number of bits for mapping them into unique integers is $L = 32$. Edge weights are equal to the algebraic complement of the relative frequency of the combination of two adjacent profiles with respect to all the observed combinations. Every graph is hashed in accordance with the procedure described in Section 3.

4.3 LSH Hyperparameters Setting

The setting provides that every network is signed by a sequence of $H = 200$ i.i.d. minhashes. Every hash is a $L = 32$ bits long integer which is a sufficient length for hashing the graphs. Each signature is grouped into B = 50 bands of R = 4 hashes combined in *bitwise XOR*. The application of the similarity criterion described in Section 3 for refining the solution only affects the subset $\mathbf{S}_3$ of the pairs in which one graph is a subgraph completely embedded into the other one. The minimum value τ for the Jaccard similarity equal to 0.491074 by assuming a 95% probability of detecting similar pairs was therefore not applied.

4.4 Some Results

The number of pairs detected is respectively equal to: $|\mathbf{S}_1| = 160370$, $\mathbf{S}_2 = 10532$ and $\mathbf{S}_3 = 242585$. The first subset contains isomorphic households which share the same number of members with the same profiles and therefore the same structure (the graphs share the same triangles). The number of distinct profiles may be equal to the number of household members at maximum. As a consequence the number of distinct triangles in common may also vary starting from a minimum value equal to 1. The second subset includes all the pairs of households with different numbers of members which share the same profiles. The third subset contains the pairs of households in which a smaller household is completely embedded into the larger one. The number of pairs in these subsets distributed by number of household members and number of common distinct profiles are reported in Table 2, Table 3 and Table 4 respectively. Except for the first subset, the number of household members reported

	1	2	3	4	5	6	7
3	50	3634	95003	0	0	0	0
4	2	38	8914	47974	0	0	0
5	0	0	119	1460	2897	0	0
6	0	0	0	10	82	180	0
7	0	0	0	0	1	4	1
8	0	0	1	0	0	0	0

Table 2: Pairs Distribution in the Subset $\mathbf{S}_1$

in the tables is always equal to the minimum value between the numbers of members of the two households in the pair. All the results reported in the tables are overall

	1	2	3	4	5	6	7
3	10	146	6126	0	0	0	0
4	1	11	288	3309	0	0	0
5	0	6	13	165	412	0	0
6	0	1	0	2	12	25	0
7	0	0	0	0	1	2	1

Table 3: Pairs Distribution in the Subset $\mathbf{S}_2$

	1	2	3	4	5	6	7
3	-	-	165200	0	0	0	0
4	-	-	9948	47916	0	0	0
5	-	-	172	2214	4644	0	0
6	-	-	3	66	249	264	0
7	-	-	2	1	13	15	4
8	-	-	0	1	2	1	1

Table 4: Pairs Distribution in the Subset S_3

counts of pairs of households and sub-households pertaining to different arrangements of the household members profiles. Although cluster analysis is not the main objective of this study, by applying the depth-search algorithm from the Graph Theory to the subsets, groups of isomorphic graphs having the same node profiles as well as groups of graphs containing the same subgraph are obtained. The results of this process are summarized in Table 5, Table 6 and Table 7 The results reported in Table 5, Table 6

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
1	2.00000	2.00000	3.00000	7.27548	7.00000	521.00000

Table 5: Clusters of Similar Graphs in the Subset S_1

and Table 7 give an idea of the diversity of the various types of networks present in the input dataset. The application of this method for the identification of groups of similar networks can lead to the detection of many small clusters containing one pair of them only. In the study presented here these results are repoted only to have an insight of the information content of the input dataset.

4.5 Accuracy of the Solution

The distributions ($D1$, $D2$) of the mean edge-weight pertaining to the left-hand side graph and to the right-hand side graph in the pairs belonging to the subsets are reported in Figure 2, Figure 3 and Figure 4: The results of the asymptotic two-sample Kolmogorov-Smirnov test are: $D = 5.612 \times 10^{-5}$ (p-value = 1) for $\mathbf{S}_1$, $D = 0.001804$ (p-value = 1) for $\mathbf{S}_2$ and $D = 0.0022013$ (p-value = 0.5992) for $\mathbf{S}_3$ where D indicates the distance between the two distributions. As can be appreciated the aforementioned

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
1	2.00000	2.00000	3.00000	2.85097	3.00000	14.00000

Table 6: Clusters of Similar Graphs in the Subset S_2

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
1	2.00000	2.00000	3.00000	27.67235	4.00000	212202.00000

Table 7: Clusters of Similar Graphs in the Subset S_3

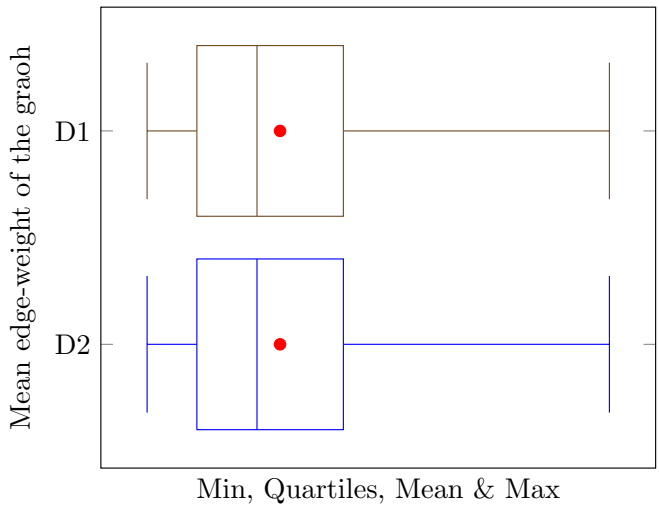

Figure 2: Distribution of the Average Edge-Weight of the Pairs Belonging to the Subset $\mathbf{S}_1$

edge-weight distributions are very close one to the other.

In order to further evaluate the accuracy of the solution, an empirical estimation of the statistical properties of hashing collisions was carried out by using the boot- strap method. In the same manner, the confidence intervals of the average number of equal bands between the signatures of similar networks, the number of equal triangles between the two graphs in each pair as well as the number of equal profiles were also calculated by bootstrapping the solutions subsets. The results are reported in Table 8, Table 9 and Table 10.

5. Conclusion

The proposed approach detects similarities between complex data structures, for exam- ple networks of individuals grouped together by any type of utility bond. By leveraging the well-known computational efficiency of the Locality Sensitive Hashing technique, the proposed approach is particularly suitable for detecting similar networks in large datasets. The use of some basic concepts from the Graph Theory offers a strong mathematical representation of these objects in that it facilitates their exploration. Networks

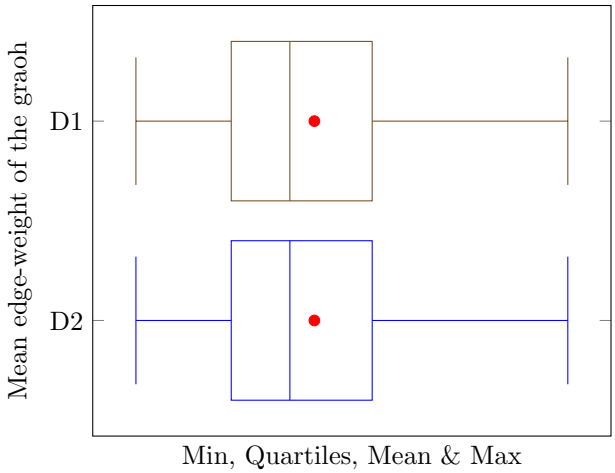

Figure 3: Distribution of the average edge-weight of tthe pairs belonging to the subset $\mathbf{S}_2$

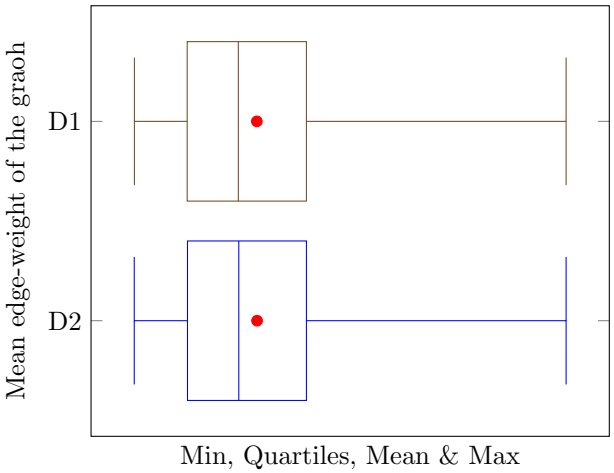

Figure 4: Distribution of the average edge-weight of tthe pairs belonging to the subset $\mathbf{S}_3$

of varying dimensions are represented as being fully connected undirected weighted graphs with attributes relating to their vertices. These attributes are comprised by a set of pre-defined categorical variables and every combination of their possible values is a profile. The weights pertaining to the edges of the graph are equal to the relative frequency of the combinations between a pair of adjacent profiles with respect to the total of the observed pairs in the input dataset. By listing all the triangles of minimum cost, every graph is transformed in a sequence of hash codes by means of an appropriate hashing algorithm. The advantage of reducing the dimensions of the problem is straightforward as the resolution of graph matching problems between all possible pairs of graphs in the input dataset turns into a more scalable search

	collisions	bands	triangles	profiles
mean	0.03471	42.50163	1.410158	3.325398
variance	0.0007791159	0.0008574714	1.816845e-06	1.989454e-06
CI lower bound (2.5%)	0	42.44742	1.407688	3.322535
CI upper bound (97.5%)	0.09	42.56104	1.412783	3.32821

Table 8: Quality of the Solution Pertaining to the Pairs in the Subset S_1

	collisions	bands	triangles	profiles
mean	0.03135	40.30934	1.460691	3.401348
variance	0.0008514275	0.01641221	3.430025e-05	3.510084e-05
CI lower bound (2.5%)	0	40.06024	1.44901	3.389278
CI upper bound (97.5%)	0.08	40.55656	1.47161	3.413124

Table 9: Quality of the Solution Pertaining to the Pairs in the Subset S_2

	collisions	bands	triangles	profiles
mean	0.03135	6.650073	1.299961	3.20174
variance	0.0003657522	0.01641221	1.076591e-06	1.323538e-06
CI lower bound (2.5%)	0	6.6136	1.298031	3.199633
CI upper bound (97.5%)	0.08	6.687575	1.302134	3.204069

Table 10: Quality of the Solution Pertaining to the Pairs in the Subset S_3

for near-duplicate graphs by approximating their Jaccard similarity index. The interesting aspect is that the proposed method addresses two types of well-known hard graph matching problems at the same time, namely the problem of finding isomorphic networks as well as the problem of detecting the common subgraph in a pair of net- works. The hashing process of the proposed algorithm is a key aspect of this study. The function proposed has satisfactory statistical properties: a long period of pseudo- random number generation and sufficient dispersion of the generated integers as the parameters of the function are chosen appropriately at random. By having selected the hashing functions uniformly at random and their modulus is large enough, the expected number of collisions is guaranteed as being low as is demonstrated in the results. The modulus adopted must also be computationally efficient in order to avoid overburdening. Nevertheless theae functions map all the different graph triangles of the reported case study adeguately. Although the probability of ensuing collisions is not null, an in-depth analysis of the solution found is required; even a samll probability of there being collisions may give rise to false positives. As a consequence, even the most accurate setting of the LSH hyperparameters is unable to avoid these collisions. The settin of the LSH hyperparameters in this study reduces the probability of there being false negatives almost to zero while this does not apply to the probability of there being false positives. The solution refinement pre-defined criterion reduces the number of detected pairs by restricting the subset of pairs of networks in which one is entirely embedded in the other only while it leaves unchanged the other subsets of isomorphic networks. The results of the in-depth analysis confirm that the detection of similar complex data structures proposed is a reliable approach.

References

1. Berg, A. C., Berg, T. L., & Malik, J. (2005, June). Shape matching and object recognition using low distortion correspondences. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)* (Vol. 1, pp. 26-33). IEEE.
2. Caelli, T., & Kosinov, S. (2004). An eigenspace projection clustering method for inexact graph matching. *IEEE transactions on pattern analysis and machine intelligence, 26*(4), 515-519.
3. Conte, D., Foggia, P., Sansone, C., & Vento, M. (2004). Thirty years of graph matching in pattern recognition. *International journal of pattern recognition and artificial intelligence, 18*(03), 265-298.
4. Liu, Z. Y., & Qiao, H. (2012, November). A convex-concave relaxation procedure based subgraph matching algorithm. In *Asian Conference on Machine Learning* (pp. 237-252). PMLR.
5. Cour, T., Srinivasan, P., & Shi, J. (2006). Balanced graph matching. *Advances in neural information processing systems, 19.*
6. Nabieva, E., Jim, K., Agarwal, A., Chazelle, B., & Singh, M. (2005). Whole-proteome prediction of protein function via graph-theoretic analysis of interaction maps. *Bioinformatics, 21*(suppl_1), i302-i310.
7. Ito, T., Chiba, T., Ozawa, R., Yoshida, M., Hattori, M., & Sakaki, Y. (2001). A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proceedings of the National Academy of Sciences, 98*(8), 4569-4574.
8. Chen, L., Vogelstein, J. T., Lyzinski, V., & Priebe, C. E. (2016, April). A joint graph inference case study: the c. elegans chemical and electrical connectomes. In *Worm* (Vol. 5, No. 2, p. e1142041). Taylor & Francis.

9. Narayanan, A., & Shmatikov, V. (2009, May). De-anonymizing social networks. In *2009 30th IEEE symposium on security and privacy* (pp. 173-187). IEEE.
10. Hu, S., Zou, L., Yu, J. X., Wang, H., & Zhao, D. (2017). Answering natural language questions by subgraph matching over knowledge graphs. *IEEE Transactions on Knowledge and Data Engineering, 30*(5), 824-837.
11. Finke, G., Burkard, R. E., & Rendl, F. (1987). Quadratic assignment problems. In *North-Holland Mathematics Studies* (Vol. 132, pp. 61-82). North-Holland.
12. Aflalo, Y., Bronstein, A., & Kimmel, R. (2015). On convex relaxation of graph isomorphism. *Proceedings of the National Academy of Sciences, 112*(10), 2942-2947.
13. Gionis, A., Indyk, P., & Motwani, R. (1999, September). Similarity search in high dimensions via hashing. In *Vldb* (Vol. 99, No. 6, pp. 518-529).
14. Bunke, H., & Shearer, K. (1998). A graph distance metric based on the maximal common subgraph. *Pattern recognition letters, 19*(3-4), 255-259.
15. Neuhaus, M., Riesen, K., & Bunke, H. (2006). Fast suboptimal algorithms for the computation of graph edit distance. In *Structural, Syntactic, and Statistical Pattern Recognition: Joint IAPR International Workshops, SSPR 2006 and SPR 2006, Hong Kong, China, August 17-19, 2006. Proceedings* (pp. 163-172). Springer Berlin Heidelberg.
16. Indyk, P., & Motwani, R. (1998, May). Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing* (pp. 604-613).
17. Shrivastava, A., & Li, P. (2014). Asymmetric LSH (ALSH) for sublinear time maximum inner product search (MIPS). *Advances in neural information processing systems, 27.*
18. Charikar, M. S. (2002, May). Similarity estimation techniques from rounding algorithms. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing* (pp. 380-388).
19. Li J., Wang J., and Wang J. (2016). Graph matching with adaptive locality sensitive hashing. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (pp. 1601-1607).
20. Achlioptas, D. (2003). Database-friendly random projections: Johnson-Lindenstrauss with binary coins. *Journal of computer and System Sciences, 66*(4), 671-687.
21. Yan X., Cheng J. and Wang J. (2018). Spherical Locality-Sensitive Hashing for Efficient Graph Similarity Search. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management (pp. 2137-2140).
22. Chen, Y., Wang, X., Bu, J., Tang, B., & Xiang, X. (2016). Network structure exploration in networks with node attributes. *Physica A: Statistical Mechanics and its Applications, 449,* 240-253.
23. Massoli, P. (2024). Detecting Similar Complex Data Structures in Large-Scale Datasets. *Curr Res Stat Math, 3*(2), 01-07.
24. Tibshirani, R. J., & Efron, B. (1993). An introduction to the bootstrap. Monographs on statistics and applied probability, 57(1), 1-436.
25. Davison, A. C., & Hinkley, D. V. (1997). Bootstrap methods and their application (No. 1). Cambridge university press.
26. Carter, J. L., & Wegman, M. N. (1977, May). Universal classes of hash functions. In Proceedings of the ninth annual ACM symposium on Theory of computing (pp. 106-112).