ISSN: 2994- 9459

Research Article

Current Research in Statistics & Mathematics

Detecting Similar Complex Data Structures in Large-Scale Datasets

Pierpaolo Massoli*

Directorate for Methodology and Statistical Process Design (DCME), Italian National Institute of Statistics (ISTAT), Via Cesare Balbo, Rome, Italy.

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

Submitted: 2024, May 06 ; **Accepted:** 2024, May 30 ; **Published:** 2024, June 11

Citation: Massoli, P. (2024). Detecting Similar Complex Data Structures in Large-Scale Datasets. *Curr Res Stat Math, 3*(2), 01-07.

Abstract

The increasing volumes of complex data stored in today's databases are driving the scientific community towards elaborating more efficient methods for data analysis. The data structures contained within them require appropriate mathematical modeling, as is the case in network structures, which can be effectively modeled by applying concepts from Graph Theory. The search for similar networks is therefore often viewed as a graph matching problem, which poses a fundamental challenge in real-world applications. This study introduces a novel approach by leveraging Locality Sensitive Hashing to efficiently address the graph matching problem. Finding an isomorphism between graphs as well as the search for the common subgraph embedded within them is achieved by hashing the graphs, thus transforming the problem into a similarity search problem. Due to its approximate nature the method applied generates false duplicates. Usual diagnostics do not guarantee high levels of accuracy of the solution. This study therefore proposes the use of the popular Conformal Prediction framework in order to evaluate the validity of the results with greater accuracy. A real-world case study is considered to test the potential of the proposed approach.

Keywords: Network Modelling, Graph Matching, Locality Sensitive Hashing, Conformal Prediction

1. Introduction

The huge 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 that of identifying similar complex data structures in largescale datasets. A widespread approach in the literature used to accomplish this task is the Graph Matching problem (GM) which searches for an alignment between the vertex sets of graphs by preserving the common structure across them. This is posed as minimizing edge disagreements over all possible vertices alignments. Graph matching has various applications in diverse fields, such as pattern recognition [1-3], machine learning [4,5], bioinformatics [6,7] neuroscience [8], social network analysis [9] and knowledge discovery in natural language processing [10]. networks can be thought of as being 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 GM problem which aims to find a bijection between the vertices of two graphs which exactly preserves 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]. Even though an extensive review of the literature pertaining to the GM problem focuses on pattern recognition topic, it is rather straightforward to accept that the graph matching can be also faced as being a similarity search problem and nearest neighbors' graphs are detected in accordance with a pre-defined metric [13-15]. Due to the fact that in a large-scale datasets applications, 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 in order to detect similar objects in high dimensional spaces by tolerating the presence of false duplicates [17-21]. In real-world applications the concept of 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 one to the other by eges or links. In mathematics, networks are often referred to as graphs so that the theoretical background of the Graph Theory cam be used for network modelling as well. One of the most important issues of network analysis is the detection of similar structures embedded in networks as is the same of determining similar subgraphs in a collection of graphs. In real world networks, nodes may have attributes which are useful for network structure exploration [22]. In this paper a large-scale dataset containing networks of different dimensions is taken into consideration. The proposed approach detects similar networks as well as sub-networks embedded into the data in accordance with an appropriate metric suitable for graph matching problems. As it is well-known from the literature in order to improve the acciracy of the results a tradeoff between false negatives and false positives is required by setting the algorithm's

hyperparameters appropriately which in real-world applications can become difficult. One of the main sources of this uncertainty relies on the hashing algorithm itself. It is impossible to avoid the hashing collisions in that they givw rise to false duplicates and uncertainty as a consequence. This study proposes to adopt the Conformal Prediction framework to evaluate the accuracy of the results [23-25].

2. Theoretical Background

In order to introduce the novel approach described in this paper 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 $i = 1,2,...,n$ vertices $v_i \in V$ and $j = 1,2,...,m$ 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. 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 k starting from vertex i and returning to the same is a sequence of distinct vertices connected by k 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* $S(G)$ is a group of fully connected vertices belonging to the vertices set of the graph.

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:

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

P∈Π

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. A general version of this problem is in general NP-hard even though in some practical applications turns into a linear assignment problem which is solvable in $O(n^3)$ for an assignment of n vertices.

2.3. Locality Sensitive Hashing Fundamentsls

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 $h(\cdot)$ 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 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:

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

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.

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 is constituted by 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 input dataset. Every vertex is related to a sequence of K categorical attributes $\mathbf{a}^{(i)} = \{a_1^{(i)}, a_2^{(i)}, \ldots, a_K^{(i)}\}$ called *profile* related to the *i-th* vertex (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 $w_{ij} \in [0,1]$

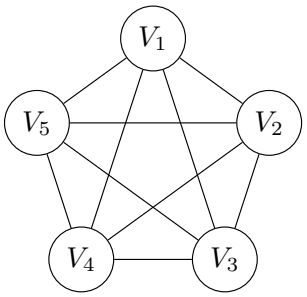

Figure 1: Input Network of N = 5 Vertices

which indicate the relative frequency of the pair of 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 respect to the attributes related to the vertices. A pair of networks which share the same node profiles corresponds to a value of the Jaccard similarity equal to 1 while this value decreases as the number of profiles in common decreases. 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. As a consequence 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 s ubgraphs.

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 $\{x_i,x_h,x_k\}$ 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 $\mathbf{t}_i(1) = x_i$, $\mathbf{t}_i(2) = x_h$ and $\mathbf{t}_i(3) = x_k$. The parameters (α_q,γ_q) are selected in order to reduce the number of collisions as much as possible. 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 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.

3.3. Optimization of The Solution

The LSH-family of minhash approximates the pairwise Jaccard similarity between the graphs. 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 insofar as a inappropriate setting could compromise the 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 by filtering out all the pairs whose Jaccard similarity is below a predefined threshold τ. This minimum acceptable similarity threshold is usually defined by inverting the Equation 4 and assuming a user-defined probability value of finding a pair of similar data. Setting this threshold has no influence on pairs of isomorphic networks and those with similarity equal to 1 but different number of vertices.

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 $|V_i| = |_{Vj}|$;
$\mathbf{S}_2$: 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 probability of there being a pair of networks with a given degree of similarity estimated as in Equation 4 does not guarantee a reliable tuning process of the LSH hyperparameters. Due to collisions caused by the hashing algorithm employed for transforming networks into signatures, false duplicates may be accidentally generated. Therefore, it is worth investigating to what extent the hashing in Equation 5 affects the accuracy of the proposed approach by means of the Conformal Prediction framework. This is a popular distribution-free technique for providing valid predictive inference for arbitrary machine learning model. It is a statistically rigorous uncertainty quantification class of methods which aims to evaluate prediction sets for classification and prediction intervals for regression depending on the type of dependent variable in the model. In this perspective the aforementioned subsets are analyzed separately. In order to evaluate the LSH model every

detected pair is considered as being an outcome of a predictive black-box model $y = f(G)$ which is considered as beung already trained by using an undefined training dataset extracted from the input data. Each pair of similar graphs act as one only so that in general is while is considered as being the true value and is the predicted value of the dependent variable for each observation i,1,2,... in the partitions of the solution. In this study the variable y is defined as being the ratio between the minimum and the maximum edge-weights of the graph OR the common subgraph in the pair. The reasoning behind this is that if there were no collisions due to to the hashing then it would be y ≡ yˆ in all cases. conformal prediction is therefore a valid strategy for quantifying the uncertainty due to the hashing collisions.

4. Application to A Statistical Population Register

The detection of complex data structures contained in statistical registers is an interesting case study for testing the potential of the proposed approach. The data source is a collection on socio-economic individual attributes describing the living conditions of a population referred to a specific time period obtained by integrating several statistical 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 specific 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 $n \geq 3$ 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, the total number of possible profiles is equal to $P = 224$.

4.2. Complex Data Structures Hashing

Representiing 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. Edge weights are

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	INCOME	Individual income (in classes)	7
5	RETIRED	Is the household member retired?	2

Table 1: Attributes in The Input Dataset

equal to the relative frequency of the two adjacent nodal profiles combination 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 $n = 32$ 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 threshold τ for refining the solution only affects the subset of the correspondences between subgraphs $\mathbf{S}_3$. By increasing the value of the threshold for the minimum acceptable similarity the dimensions of the detected common subgraph increase even if the number of the involved pairs decrease. In this case the minimum value τ for the Jaccard similarity is equal to 0.491074 by assuming a 95% probability of detecting similar pairs.

4.4. Some Results

In order to better investigate the results the three partitions of the solution set are analyzed separately. The number of pairs detected is respectively equal to: $|\mathbf{S}_1| = 254227$, $|\mathbf{S}_2| = 29870$ and $|\mathbf{S}_3| = 394627$. The first subset contains isomorphic households which share the same number of members with the same profiles and therefore the same structure. The number of distinct profiles may be equal to the number of household members at maximum. 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 which share a common *nucleus* embedded in the households of the pair. 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 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 table are overall counts of pairs of housenolds and nucleus pertaining to different arrangements of the household members profiles.

4.5. Uncertainty Quantification

On the basis of the reasoning described in Section 3 every subset of the solution was split into a *calibration* set (80%) and a test set (20%). The uncertainty was evaluated

	1	2	3	4	5	6
3	29	6684	157708	0	0	0
4	1	55	21651	61368	0	0
5	0	0	387	3247	2628	0
6	0	0	4	86	188	175
7	0	0	0	4	3	6
8	0	0	1	0	2	0

Table 2: Pairs Distribution in The Subset S1

	1	2	3	4	5	6	7
3	14	370	15445	0	0	0	0
4	0	7	2586	9166	0	0	0
5	0	1	79	973	1001	0	0
6	0	0	1	48	75	79	0
7	0	0	0	4	11	4	2
8	0	0	0	0	1	2	0
9	0	0	0	0	0	1	0

Table 3: Pairs Distribution in The Subset S2

	3	4	5	6	7	8
3	244560	0	0	0	0	0
4	30897	63008	0	0	0	0
5	1074	6783	7048	0	0	0
6	53	384	753	379	0	0
7	2	42	85	77	11	0
8	1	5	30	15	6	0
9	0	2	10	3	1	0
10	0	0	1	1	0	0
11	0	0	0	0	2	0

Table 4: Pairs Distribution in The Subset S3

by using the standardized scores defined as follows:

$$s(x,y) = \frac{|y - \hat{y}|}{\hat{\sigma}} \qquad (6)$$

where $\hat{\sigma}$ is the the standard deviation of the predicted variable $\hat{y} = f(x)$ estimated for each observation $x = G$ in the calibration dataset. As it is well-known the prediction interval for each observation in the test dataset is equal to $C(x) = [\hat{f}(x) - \hat{\sigma}\,\hat{q}, \hat{f}(x) + \hat{\sigma}\,\hat{q}]$ where $\hat{q}$ is the $\lceil (1-\alpha)(n+1) \rceil / n$ quantile with n equal to the number of observations in the calibration dataset and $\alpha = 0.05$ is the user-defined level of accuracy. The probabilitiy that the true value y falls into the prediction interval is as follows:

$$P[s(x,y) \leq \hat{q}] \geq 1 - \alpha \qquad (7)$$

where the scores are computed for each observarion belonging to the test dataset. This is the property to assess in order to test for the accuracy of the LSH detection in the proposed approach. The correctness of this *coverage* property was checked out by using the efficient score caching proposed by Angelopoulos and Bates. The resulting coverage is approximately equal to 95.05% in the case of the pairs belonging to $\mathbf{S}_1$ subset, 95.36% in the case of the $\mathbf{S}_2$ subset and 95.04% in the case of the $\mathbf{S}_3$ subset.

5. Conclusion

The proposed approach identifies similarities between complex data structures, for example 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 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 patrs 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 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 networks. Due to its approximate nature even the most accurate setting of the hyperparameters is not sufficient to avoid the presence of false duplicates. The setting proposed 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 even if the rate of the same is reduced by usinng a pre-defined threshold of the minimum acceptable Jaccard similarity. As it is known from the literature, hashing gives rise to inevitable collisions which may generate false duplicates which are not avidable even with the most accurate tuning process of the hyperparameters. As a consequence, the standard disgnostic of the LSH tecnique is not reliable. On the basis of this reasoning, the application of the Conformal Prediction framework is employed for evaluating the accuracy of the results. The case study indicates that the proposed approach is computationally efficient as well as accurate as it is demonstrated by applying a simple but rigorous uncertainty quantification method of the Conformal Prediction.

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., Praveen S., and Jianbo S. (2007). Balanced Graph Matching. Edited by B. Scholkopf, J. C. Platt and T. Hoffman, 313-20. http://papers.nips.cc/paper/2960balanced-graph-matching.pdf.
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. Vovk, V., Gammerman, A., & Shafer, G. (2005). *Algorithmic learning in a random world* (Vol. 29). New York: Springer.

24. Shafer, G., & Vovk, V. (2008). A tutorial on conformal prediction. *Journal of Machine Learning Research, 9*(3).
25. Angelopoulos, A. N., & Bates, S. (2021). A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv preprint arXiv:2107.07511*.