Active Learning for the Subgraph Matching Problem

Yurun Ge, Andrea L. Bertozzi
Department of Mathematics
University of California, Los Angeles
Los Angeles, CA 90095

2021 IEEE International Conference on Big Data (Big Data) | 978-1-6654-3902-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/BigData52589.2021.9671760

***Abstract*—The subgraph matching problem arises in a number of modern machine learning applications including segmented images and meshes of 3D objects for pattern recognition, biochemical reactions and security applications. This graph-based problem can have a very large and complex solution space especially when the world graph has many more nodes and edges than the template. In a real use-case scenario, analysts may need to query additional information about template nodes or world nodes to reduce the problem size and the solution space. Currently, this query process is done by hand, based on the personal experience of analysts. By analogy to the well-known active learning problem in machine learning classification problems, we present a machine-based active learning problem for the subgraph match problem in which the machine suggests optimal template target nodes that would be most likely to reduce the solution space when it is otherwise overly large and complex. The humans in the loop can then include additional information about those target nodes. We present some case studies for both synthetic and real world datasets for multichannel subgraph matching.**

I. Active Learning

Active learning is an area of research in statistical machine learning that brings a subject matter expert (SME) into the actual algorithm for classification of points in a dataset. Supervised machine learning algorithms involve an abundance of labels. In the real-world however, unlabeled data is common and accurate labeling may require human involvement that can not be crowd-source due to privacy or security reasons. Semi-supervised methods use significantly fewer training points. At the same time, the choice of labelled data often affects classifier performance. Active learning involves the use of an algorithm or formula to choose individual data points for labeling by a SME, the results of which are then included in a semi-supervised learning problem. This procedure can be iterated sequentially or in batch. These methods iterate between: (1) training a model given the current labeled data (2) choosing a set of active learning query points in the unlabeled set according to an acquisition function (also called an active learning criterion). Most active learning acquisition functions for statistical belong to one of a few categories: uncertainty [42], [21], [14], margin [47], [2], [22], clustering [13], [28], and look-ahead [56], [5]. Fig 1 shows a diagram of active learning in machine classification. Perhaps a more relevant delineation is between sequential active learning and batch active learning. In the sequential case, one unlabelled node at a time is given to the human in the loop to label, whereas in batch learning, a batch of nodes are processed together. This distinction is also relevant for active learning for subgraph matching.

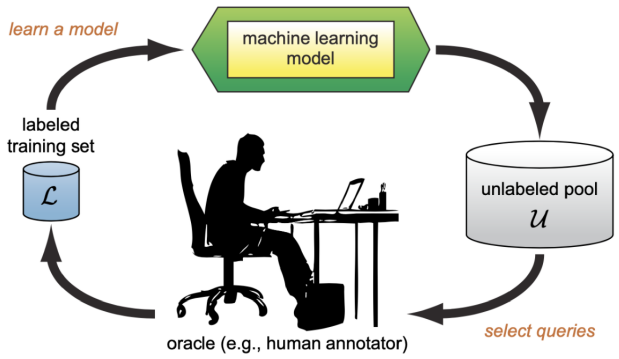

Fig. 1: Active learning flowchart for statistical machine learning classifier from [42]. The selected queries are often determined by an acquisition function.

Researchers are interested in introducing active learning into the network alignment problem, which tries to find an optimal mapping of graph nodes with maximum similarity between the nodes and edges. In the network alignment problem, usually a cost function is defined to measure the difference between the nodes and edges. The optimal solution with least cost is given by updating the probability distribution for each node. Previous research shows that better alignment can be achieved by introducing interaction with a human to obtain extra information on certain nodes. For example, in [41], researchers compare three probability matrix based query strategies. In [10], active learning is introduced after a machine learning on the cost function. In [30], instead of asking for exact information on certain nodes, the research examines more ambiguous query questions.

Email: yurun97@ucla.edu, bertozzi@math.ucla.edu, This work was partially supported by the AirForce Research Laboratory and DARPA under agreement number FA8750-18-2-0066. This work was also supported by NSF grant DMS-2027277. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Research Laboratory and DARPA or the U.S. Government.

Active learning algorithms have not been seriously studied for subgraph isomorphisms however their need is justified by the complexity of the solution space common for such problems. First we review the multiplex network defined in [36].

Definition 1 (Multiplex Network)**.** *A multiplex network $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{L}, \mathcal{C})$ is a set of nodes (frequently called vertices), directed edges between the nodes, labels on the nodes, and channels on the edges. The number of nodes is denoted n. Each node $\mathbf{v} \in \mathcal{V}$ has a label $\mathcal{L}(\mathbf{v})$ belonging to some arbitrary set of labels. There can be any number of edges between each pair of nodes $(\mathbf{u}, \mathbf{v})$ in either direction. Each edge belongs to one of the channels $\mathcal{C}$. Edges between the same pair of nodes in the same channel with the same direction are indistinguishable. The function $\mathcal{E} : \mathcal{V} \times \mathcal{V} \to \mathbb{N}^{|\mathcal{C}|}$ describes the number of edges in each channel between each pair of nodes. In particular, $\mathcal{E}(\mathbf{u}, \mathbf{v})$ can be represented as a $|\mathcal{C}|$-dimensional vector the k^{th} element of which is the number of edges from node $\mathbf{u}$ to node $\mathbf{v}$ in the k^{th} channel. $|\mathcal{E}|_0$ denotes the number of distinguishable edges in $\mathcal{G}$.*

The subgraph matching problem can be succinctly stated: Given two multiplex networks, a template $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t, \mathcal{L}_t, \mathcal{C})$ and a world $\mathcal{G}_w = (\mathcal{V}_w, \mathcal{E}_w, \mathcal{L}_w, \mathcal{C})$, we explore the space of all subgraphs of the world that *match* the template. There are several closely related problems with different computational costs. Each of these problems relies on the same concept of a multiplex *subgraph isomorphism (SI)* as described in [36].

Definition 2 (SI: Subgraph Isomorphism)**.** *An injective function $f : \mathcal{V}_t \to \mathcal{V}_w$ is called a* subgraph isomorphism (SI) *from $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t, \mathcal{L}_t, \mathcal{C})$ to $\mathcal{G}_w = (\mathcal{V}_w, \mathcal{E}_w, \mathcal{L}_w, \mathcal{C})$ if*

$$\begin{aligned} \mathcal{L}_t(\mathbf{v}) &= \mathcal{L}_w(f(\mathbf{v})) && \forall \mathbf{v} \in \mathcal{V}_t \\ \mathcal{E}_t(\mathbf{u}, \mathbf{v}) &\leq \mathcal{E}_w\left(f(\mathbf{u}), f(\mathbf{v})\right) && \forall \mathbf{u}, \mathbf{v} \in \mathcal{V}_t \times \mathcal{V}_t. \end{aligned}$$

The set of all SIs from $\mathcal{G}_t$ to $\mathcal{G}_w$ is denoted $\mathcal{F}(\mathcal{G}_t, \mathcal{G}_w)$.

This definition allows for isomorphisms in which the world graph has more edges than the template. An *induced subgraph* is a special case in which the edge count in the template and world are the same for those nodes in the template and its image. Despite this very simple definition, real world, synthetic, and benchmark examples illustrate the complexity of the solution space. For example, for single channel (single edge-type) networks, there are several benchmark datasets [44], [12], [45], [17], [32] for which the total SI count ranges from zero to 10^{384}, approximately [53]. The need for a SME/analyst team for multiplex subgraph matching is illustrated quite well by the benchmark datasets developed under the DARPA MAA program (Modeling Adversarial Activity) [40]. The theme of this program involves template graphs that describe a series of actions and the world graph is constructed from relevant data. We review several examples from the MAA program along with a transportation example from the public domain. We describe how an active learning scenario could be implemented in the context of constraint propagation algorithms for subgraph matching. A flowchart describing how this approach might be used in a real world setting in shown in Fig. 2. In this paper the subgraph matching algorithm in the flowchart is one of constraint propagation. It determines potential candidate world nodes to match to each template node. In the next section we review algorithms for subgraph matching and discuss how the active learning framework is combined with subgraph search strategies.

II. Algorithms for subgraph matching

Most algorithms for subgraph isomorphisms use one of three approaches [6], [7]: tree search, constraint propagation, and graph indexing. Tree search is one way to find subgraph isomorphisms. The bookkeeping keeps track of a search state while navigating the tree of possible search states, backtracking when reaching the end of a branch. This approach has considerable computational complexity and thus refinement of the search space is needed to avoid unnecessary branches. Tree search methods include Ullmann's algorithm [50], VF2 [11] and its variants (VF2Plus [8], VF3 [6], [7], VF2++ [23]), and for specific graphs, RI/RI-DS [4].

Constraint propagation approaches cast the problem as a constraint satisfaction problem. One keeps a record of world nodes that are possible matches for each template node. By repeatedly applying local constraints, the candidate list is reduced until only a few possible matches remain. This approach can be combined with a tree search to solve the subgraph matching problem. Examples of constraint propagation approaches include McGregor [34], nRF+ [27], ILF [54], LAD [43] (and its variants, IncompleteLAD and PathLAD [26]), McCreesh and Prosser (Glasgow) [31], and FocusSearch [51]. A multiplex subgraph matching code was introduced in [35], [36] for multichannel/multi-edge templates and world graphs including some of the DARPA MAA datasets. These are the class of methods that we explore for the active learning problem. The primary code we used for filtering is a multichannel adaptation of the Glasgow solver [33], detailed in [53].

The subgraph matching problem includes several different levels of solutions for subgraph isomorphisms, enumerated in Table I - the subgraph isomorphism problem (SIP), the signal node set problem (SNSP), the minimum candidate set problem (MCSP), the subgraph isomorphism counting problem (SICP), and the subgraph matching problem (SMP).

Problem	Description
SIP	Check if there are *any* SIs.
SNSP	Find all the world nodes involved in SIs.
MCSP	Find all pairs $(\mathbf{u}, \mathbf{v})$ where $\mathbf{u} = f(\mathbf{v})$ for some SI f.
SICP	Count the number of SIs.
SMP	Find all the SIs.

TABLE I: A summary of the various problems for subgraph isomorphisms, in increasing order of computation cost [36].

The subgraph matching problem is combinatorially complex. This is largely due to two features of the problem (a) that the world graph has additional nodes and edges that

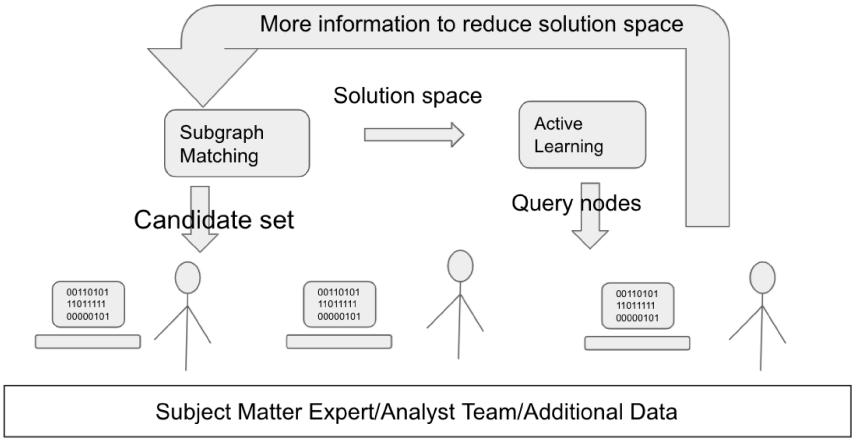

Fig. 2: Active learning flowchart for subgraph matching. A subgraph matching algorithm determines all potential candidates for template nodes (using constraint propagation). An active learning algorithm determines the optimal nodes for SMEs to obtain additional constraints/information. This is fed back into the subgraph matching algorithm.

are equally good candidates for components of the template graph and (b) the template has nodes and edges that are interchangeable. These are forms of equivalence that have been explored recently in the literature [49], [53] to reduce the complexity of the solution space by providing a categorization of groups of nodes that can be interchanged.

The problem we are interested in is to solve the SMP while simultaneously ruling out SIs that can be eliminated by additional information that is available or potentially available to SMEs and analysts. The end goal is to have a final solution to the SMP, after elimination of extraneous SIs, that has a modest solution count and provides a final list of SIs that are clearly of interest to the the application problem. The SMEs are part of the active learning procedure, providing information based on active learning queries. We propose some strategies for these queries in the next section.

III. Active learning for subgraph matching

One might have the objective to identify one specific subgraph isormorphism by restricting the candidate nodes/edges in the world graph. There are a number of reasons why this would be - for example if the knowledge graph represents data related to an investigation involving an unknown actor, such as in a homicide investigation or a serial offender, it would be important to identify the actual person involved. The consequences of misidentifying someone could be grave - both for the person wrongly identified and for potential future victims of the actual person involved. In some cases there could be more than one subgraph isomorphism of relevance, for example in the case of identifying different but equally important pathways in a biochemical reaction network or the case of identifying groups involved in human trafficking or smuggling. Likewise, organizations or people interested in identifying those wrongly accused of crimes could look at a knowledge graph of information that might present alternate scenarios. In a real life setting, this could entail addition additional constraints added to the problem space such as attributes for the nodes (e.g. names, dates, times etc). It could also involve addition of more data. Such information might come at a cost and therefore it would be of interest to understand strategies to reduce the complexity of the solution space with the minimal cost.

Here we look at some examples of multichannel networks in which the solution space is combinatorially large. We look at the probablity of fixing the candidate nodes for a small number template nodes and we ask the question - which template nodes should be chosen so as to reduce the complexity of the solution space the most?

Below we propose a few simple querying strategies for active learning to reduce the solution space. The querying strategies are carried out after first running the constraint propagation algorithm to determine a potential list of candidate nodes. In numerical examples in this paper we choose simple filtering strategies without extensive tree searches. Thus we are not solving the SNSP or MCSP in full, rather providing a pared down list of candidates under consideration for the subgraph matching problem. The reason for this is that an active learning method requires code that can run in real time for analysts and this will be essentially guaranteed for the constraint filters but not for extensive tree searches to validate all the candidates.

IV. Querying strategies for template nodes

This paper focuses on querying strategies for template nodes. These are the easiest to analyze and visualize by displaying the candidate counts for each template node, with the templates being small enough that they can be displayed simply in a two dimensional diagram. Such a strategy is also important for SME/analysts to interact with the active learning algorithms. Below we present several strategies for choosing template nodes to query.

A. Local template-based strategies

First we consider two simple strategies:

- choose the template nodes with the largest degree centrality measure (number of edges connecting that node)
- choose the nodes with largest sum of the number of candidates for neighboring template nodes

B. Edge entropy

We introduce a notion of "edge entropy". One purpose of the query is to simplify the complex part of the graph to enable less costly tree searches. Shannon's entropy is one tool to measure complexity. In the subgraph matching problem, the mapping of an edge is usually more complex than the mapping of a node. We define the following edge entropy:

$$-\sum_i p_i log(p_i), \tag{1}$$

where p_i is the probability that the mapping of an edge to its candidate set passes the local filter in the affected region. Here i is summed over all edges connected to the node in question and the entropy measure is assigned to that node.

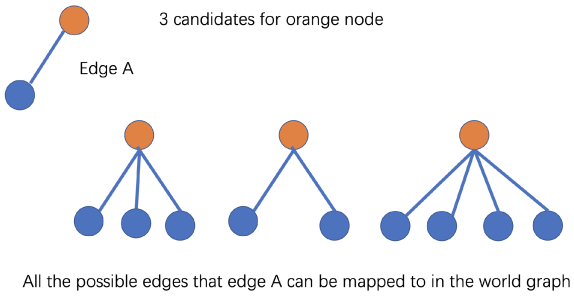

Fig. 3: Edge entropy toy example

Fig. 3 shows a toy example, in which we want to query information for the orange node, connected to edge A. In the world graph, this edge can be mapped to nine possible candidate edges. There are three cases that the orange node can map to. In the first case, there are three edges connected to the selected world node. So the probability in this case is $1/3$. Similarly the probability for the remaining edges are $2/9$ and $4/9$ So the edge entropy of this edge is $-(\frac{1}{3}log(\frac{1}{3}) + \frac{2}{9}log(\frac{2}{9}) + \frac{4}{9}log(\frac{4}{9}))$. For each template node, we calculate the sum of edge entropies of all the edges that are incident to the template node.

While pruning the candidate list, we may run into cases where the size of the candidate set is too large. In this case, we are unlikely to find all the subgraph isomorphisms using tree search because of limited computation time and resources. By introducing active learning in the pruning of candidate list, we can query information about a certain nodes or edges in the template or world graph to reduce the candidate list to an acceptable size. The problem of determining the nodes and edges to query is the active learning problem in subgraph isomorphism problem. Below we show some examples using datasets for multichannel networks.

V. IVYSYS V7 - SUM OF CANDIDATES

We show an example from IvySys Version 7 [1], developed by Ivysys technologies for the DARPA MAA program, with three channels corresponding to financial, communication and logistics transactions. This dataset has a template with 92 nodes and 195 edges. The world graph has 2,488 nodes and 5,470,970 edges. To date the entire solution space has not been solved for, although a representative solution with over 10^{100} isomorphisms is identified in [53]. The template has a tree-like sparse structure, resulting in no unique candidates after applying different levels of filtering methods, as seen in Fig. 4 (left).

We can significantly reduce the solution space by querying key nodes, selected according to the maximum sum of neighboring candidates (see Fig. 4 (right - the nodes are dark green)). We note that the degree centrality metric identifies five out of six of the same query nodes in this example. The edge entropy criterion identifies the same six nodes but with a different ordering. The sixth one is the query node that does not have many leaves. Based on the query from the active learning criteria, we specific world nodes to the queried template nodes. We chose world nodes from the first isomorphism found by the code from [53], as a proxy for additional information supplied by SMEs. After fixing the queried nodes, the remaining candidates in the center of the template are reduced significantly, mostly to a single world node or a few world nodes. This example has a network structure reminiscent of core-periphery structure [39]. However, the leaf nodes connecting to the core nodes still have many candidates. Due to large equivalence classes in the template and world graph [53], [37], the number of solutions for the SMP problem is still huge. That said, one can still obtain useful information for SMEs and analysts with a Venn diagram representation of candidates for several equivalence classes as shown in Fig. 4 in the bottom row. It shows the intersection of candidate sets of the largest three equivalent classes in the template. The difficulty of the subgraph isomorphism comes from the alldifferent problem [38] in assigning the template nodes to its candidate set. But analysts can get an idea of what the solution space looks like before solving the all-different problem. The Venn Diagram itself could be incorporated into a computational tool in which an analysts could click on a portion of the Venn diagram to obtain an itemized list of those candidates.

VI. EXAMPLE FROM PNNL REAL WORLD

This dataset was made by Pacific Northwest Nationabl Lab from a social media dataset collected by Matteo Magnani and Luca Rossi [9], [29]. It involves friend/follower relationships on three social media platforms, each of which corresponds to a channel. The template graph has 35 nodes and 158 edges and is an *induced subgraph* in the world graph.The world graph has 6,407 nodes and 74,862 edges in six channels. Additional SI solutions can be found with additional edges. We use the induced subgraph as "ground truth" for our query analysis. We refer to this dataset and its induced subgraph template as

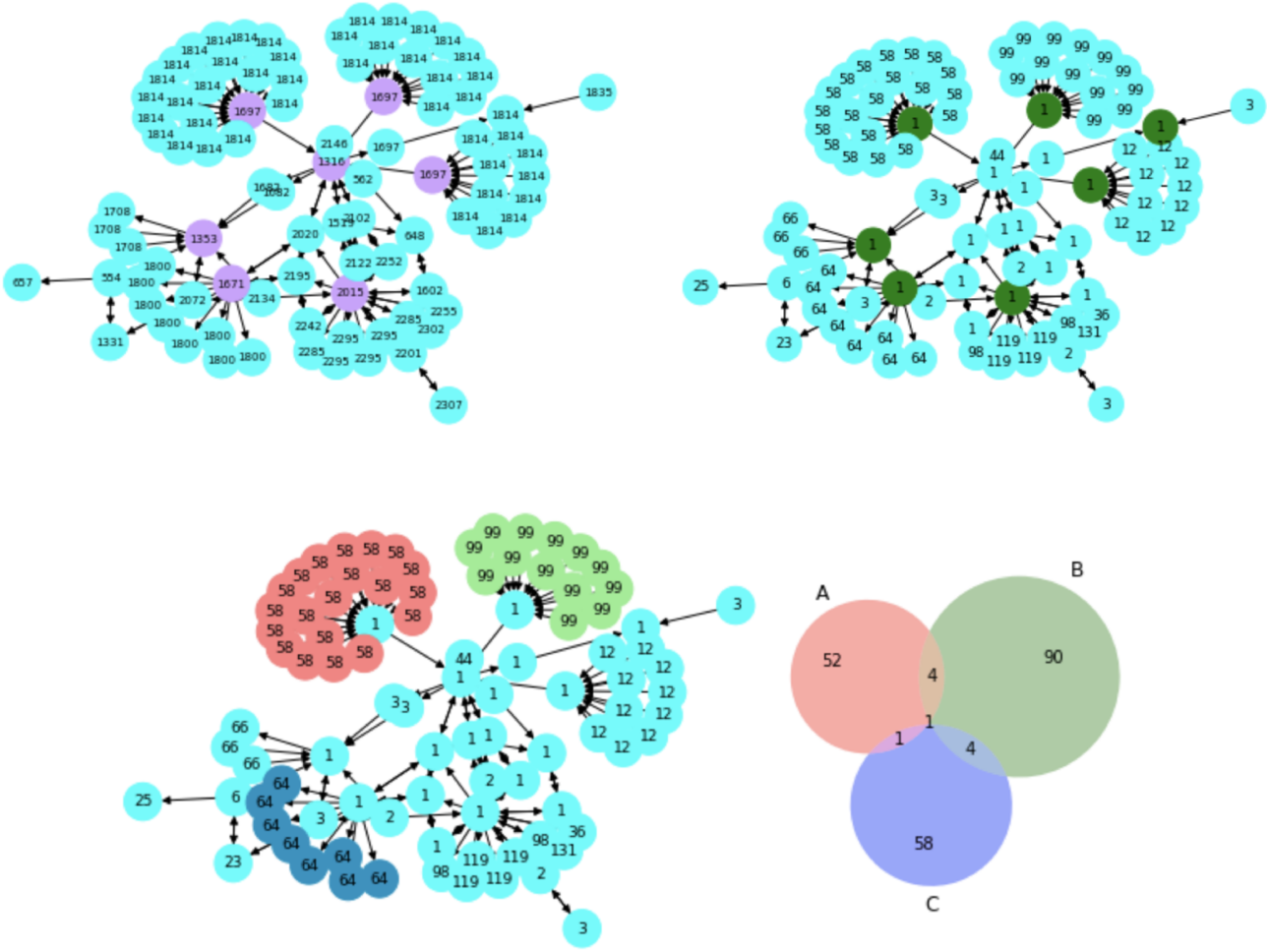

Fig. 4: (Top left) Number of candidates for querying the nodes of ivyvys v7 template after implementing the main filters in [53]. The template nodes with the highest degree centrality are marked in purple. (Top right) Number of candidates after querying the template nodes with maximum sum of neighboring candidates, using the first isomorphism in the first found representative solution using the code in [53]. (bottom) The overlapping structure of candidate sets after the query. The circles in the Venn diagram represent the candidate sets of the nodes in the template with corresponding color - Set A (red nodes), Set B (green nodes), Set C blue nodes).

the "PNNL Real World dataset". An analysis in [36] found a total of 2.12×10^{12} isomorphisms using all filters including the elimination filter which performs a final tree search. Fig. 5 shows the template for this subgraph matching problem in which each node has listed the number of candidates from the world graph after applying the node level statistics, topology, repeated-set, and neighborhood filters (but not elimination filter) [36]. For this reason the candidate counts are slightly higher than what are shown in [36] and are more realistic for an active learning scenario when there may not be sufficient time to run extensive tree searches, especially when additional information can be added to greatly reduce the solution space. The entropy values of each of the template nodes are shown in the top right. The full candidate count is shown in the bottom figures for two choices of queries - on the left the top two entropy node are chosen. On the right the third and fourth highest entropy nodes are chosen. The choice of the highest entropy nodes clearly has a much smaller solution space than the resulting solution space for the alternate choices. We also tried the sum of neighboring candidate counts as a metric and found that we needed to query the top three nodes with that metric to get the same reduction of the solution space as what was found by querying the top two nodes according to the entropy metric.

In summary, For PNNL real world, the two nodes selected by the sum of candidates for neighboring nodes also have highest entropy. However the nodes with highest degree centrality are different and only have one candidate each after applying the basic filters. Selecting nodes with higher entropy performs much better than selecting nodes with lower entropy.

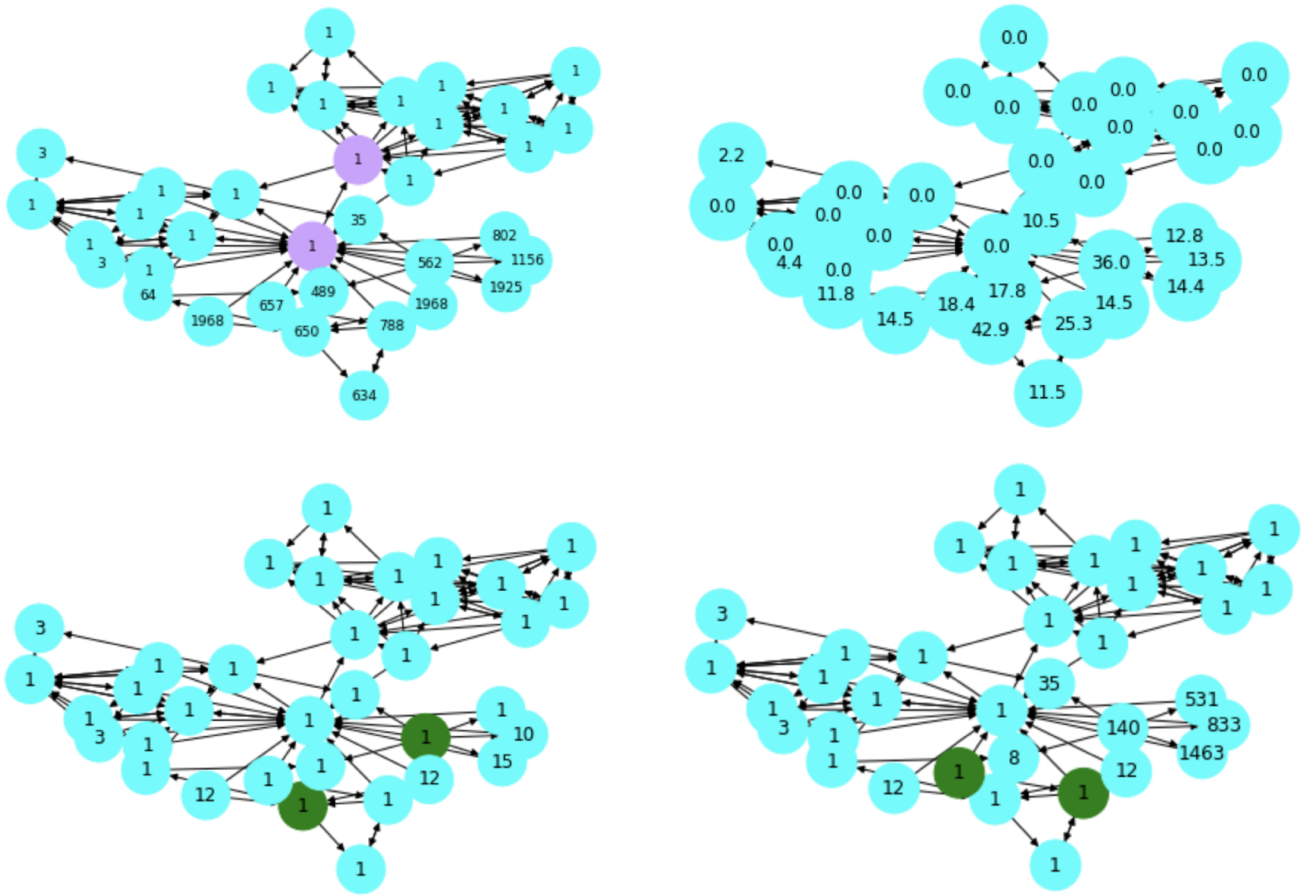

Fig. 5: (Top left) Number of candidates of PNNL real world template, after running basic filters. The template nodes with the highest degree centrality are marked in purple. Notice that they each only have one candidate node and are thus not useful to query in an active learning scenario. (Top right) Entropy values for each each template node. (bottom left) Number of candidates after querying the two nodes with highest entropy. (Bottom right) Number of candidates after querying the the nodes with third and fourth highest entropy rather than the top two highest entropy.

VII. Example from Great Britain Transportation Network

The Great Britain Transportation Network [15] is comprised of the public transportation dataset available through the United Kingdom open-data program [52] with timetables of domestic flights in the UK. It is a multiplex time-dependent network. There are six channels involving different transportation methods, including bus, air, ferry, railway, metro, coach. This dataset has 262,377 nodes and 475,502 edges. The original dataset can be found at [16]. The authors of [36] have an online interactive map [20] for users to visualize the template. We use the template graph constructed in [36]. The authors identified a small set of locations that interact with each other through all channels (excluding airlines, since this channel is very sparse). If a location involves all five non-air channels in the network, we assume that it is important. There are only three nodes that interact in the 5 non-air channels, and they randomly chose one of them as the template center, specifically the Blackfriars Station in London. Starting from this node,they used a random walk to create a template with 53 nodes and 56 edges. As in the previous example the template comes from an *induced subgraph* in the world network which serves as the "ground truth" for the active learning problem.

After iteratively applying filters, the candidate number for each node is large for every template node, which would result in a long time for a tree search to solve the MCSP or the SMP. Considering the cities each node represent in real life, the subgraph isomorphism problem only has one "correct" solution. So acquiring additional information is necessary. The template can be divided into two parts. On the left and bottom part there are two tree-like tails. The rest of the graph is the core part which is relatively dense. In the tail part, we determine that a node in the middle of the tail is the optimal choice however it does not have maximal entropy nor maximal sum of neighboring candidates. This is shown in Fig 6 top right. The same figure on the bottom shows results for max sum of candidate nodes vs. max entropy queries. The max sum of candidate nodes outperforms the max entropy metric

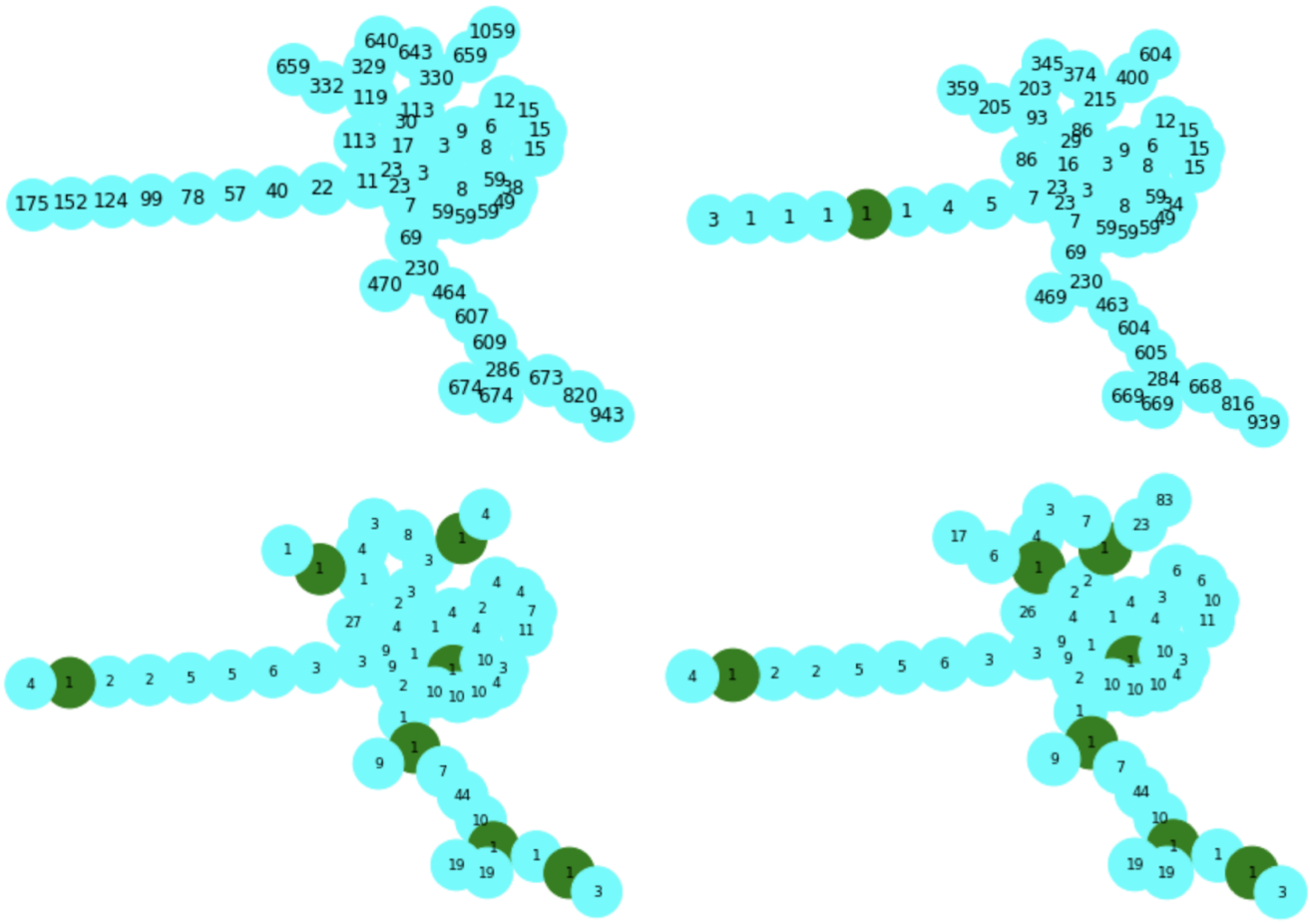

Fig. 6: (Top left) Number of candidate nodes for the British Transportation Network template from [36]. (Top right) Number of candidates after querying the middle node of the tail. (Bottom left) seven query nodes chosen according to max sum of candidates for neighbors. (bottom right) seven query nodes chosen according to max entropy.

for this example. The optimal choice would actually take the entropy nodes and replace the node towards the end of the long tail with the middle node in the top right panel. We found this node by trial and error and this suggests that there are going to be some examples for which other metrics are needed to optimize the choice of query nodes. This is also still an open and active problem for statistical machine learning as well.

VIII. Conclusions and Future Work

The subgraph matching problem is usually related to concrete real life problems. Graphs created from such problems typically have a combinatorially complex solution space. Even with edge and node attributes, the solution space can be large [49]. We have defined a methodology for introducing active learning to the subgraph matching problem. It involved a feedback mechanism between a computer supplying information and suggesting nodes to query for maximal benefit, and SMEs and analysts introducing additional information to reduce the problem space. In this work we introduced several strategies for querying template nodes and discussed their performance on two datasets from the DARPA MAA program and from one public dataset involving the transportation network of Great Britian. The examples showed some promise for using measures such as the sum of candidates of neighboring template nodes and an edge entropy measure to optimize the use of information to reduce the solution space. Our examples also show that the degree centrality may not be as useful for identifying query nodes in the template. For the Ivysys V7 example, the sum of neighboring candidate metric was explored with six query nodes. The edge entropy criteria found the same six nodes but with a different ordering. The centrality measure found five of the six nodes. For the PNNL real world example the edge entropy metric outperformed the sum of neighboring candidates metric. We also show that for the Great Britain Transportation Network, which involves long chains that are commonplace in ground transportation networks (subway and bus lines for example), that the methods that work well in the previous examples are less helpful for the long tail part of this structure. We find that the max candidate metric outperforms the max entropy metric for this example. The examples shown here suggest that the optimal choice of

template nodes to query depends heavily on the graph topology and structure and is an interesting problem for future research. The calculations done here were performed with sequential queries although these could also be done in batch processing. The distinction between those cases is also an interesting topic for further study.

An important future direction involves approaches for querying the world graph nodes instead of the template nodes. Two possible strategies include:

- querying the world node with highest degree.
- querying the world nodes that are the candidates of the most template nodes.

Different querying options will have different costs in terms of availability of data and time required to obtain additional data. One might expect that querying a template node could have higher cost than querying on the world node, however this may depend on the application.

There are a number of algorithms and problems not considered here that are interesting choices for active learning methods. For example, for the inexact subgraph match problem [49], [25], [46], one can develop a similar strategy with additional metrics for the closeness of the graph match. Another variant on the subgraph matching problem are pathway identification in graphs [48]. In addition, structural equivalence [53], [37] uses symmetries in the template and world graph to reduce the complexity of the solution space and these equivalences will also aid in the active learning problem. The Ivysys example in this paper exhibits significant structural equivalence. The use of such information can sometimes lead to the difference between solving the SICP vs not solving it.

Finally we remark that the active learning strategies proposed are very simple - they are based on easily computable metrics. In the stastical machine learning setting, an active area of research involves "look ahead" models that leverage the classifier's state to "look ahead" at what changes would occur in the as a result of labeling an unlabeled point - references include the seminal work of Zhu et al [56] as well as the EMCM [5] and Maxi-Min "data-based norm" [24] methods. For the subgraph matching problem, tree search methods and additional constraint propagation methods could be part of a look-ahead approach to further optimize active learning queries, looking beyond graph neighborhood statistics. Graph indexing approaches to the subgraph matching problem including GraphQL [19], SPath [55], TurboISO [18], and CFL-Match [3], use various pattern matching approaches, including nonlocal ones, to solve the SMP. Their query-based format may provide a useful structure for active learning.

ACKNOWLEDGMENTS

We thank Dominic Yang for useful conversations regarding structural equivalence. We thank Kevin Miller for helpful conversations regarding active learning for statistical machine learning. We thank the authors of [53] for use of their code.

REFERENCES

[1] K. O. Babalola, O. B. Jennings, E. Urdiales, and J. A. DeBardelaben. Statistical methods for generating synthetic email data sets. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 3986–3990, Dec 2018.

[2] Maria-Florina Balcan, Alina Beygelzimer, and John Langford. Agnostic active learning. In *Proceedings of the 23rd international conference on Machine learning*, ICML '06, pages 65–72, Pittsburgh, Pennsylvania, USA, June 2006. Association for Computing Machinery.

[3] F. Bi, L. Chang, X. Lin, L. Qin, and W. Zhang. Efficient subgraph matching by postponing cartesian products. In *Proceedings of the 2016 International Conference on Management of Data*, pages 1199–1214. ACM, 2016.

[4] V. Bonnici, R. Giugno, A. Pulvirenti, D. Shasha, and A. Ferro. A subgraph isomorphism algorithm and its application to biochemical data. *BMC bioinformatics*, 14(7):S13, 2013.

[5] Wenbin Cai, Ya Zhang, and Jun Zhou. Maximizing Expected Model Change for Active Learning in Regression. In *2013 IEEE 13th International Conference on Data Mining*, pages 51–60, December 2013. ISSN: 2374-8486.

[6] V. Carletti, P. Foggia, A. Saggese, and M. Vento. Introducing VF3: A new algorithm for subgraph isomorphism. *Graph-Based Representations in Pattern Recognition*, pages 128–139, 2017.

[7] V. Carletti, P. Foggia, A. Saggese, and M. Vento. Challenging the time complexity of exact subgraph isomorphism for huge and dense graphs with vf3. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):804–818, April 2018.

[8] V. Carletti, P. Foggia, and M. Vento. Vf2 plus: An improved version of vf2 for biological graphs. In *International Workshop on Graph-Based Representations in Pattern Recognition*, pages 168–177. Springer, 2015.

[9] F. Celli, F. M. L. Di Lascio, M. Magnani, B. Pacelli, and L. Rossi. Social Network Data and Practices: the case of Friendfeed. In *International Conference on Social Computing, Behavioral Modeling and Prediction*, Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2010.

[10] Donatello Conte and Francesc Serratosa. Interactive online learning for graph matching using active strategies. *Knowledge-Based Systems*, 205:106275, 2020.

[11] L. P. Cordella, P. Foggia, C. Sansone, and M. Vento. A (sub)graph isomorphism algorithm for matching large graphs. *IEEE Trans. on Pattern Analysis and Machine Intell.*, 26(10):1367–1372, Oct 2004.

[12] Guillaume Damiand, Christine Solnon, Colin De la Higuera, Jean-Christophe Janodet, and Émilie Samuel. Polynomial algorithms for subisomorphism of nd open combinatorial maps. *Computer Vision and Image Understanding*, 115(7):996–1010, 2011.

[13] Sanjoy Dasgupta and Daniel Hsu. Hierarchical sampling for active learning. In *Proceedings of the 25th international conference on Machine learning*, ICML '08, pages 208–215, Helsinki, Finland, July 2008. Association for Computing Machinery.

[14] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep Bayesian active learning with image data. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pages 1183–1192, Sydney, NSW, Australia, August 2017. JMLR.org.

[15] R. Gallotti and M. Barthelemy. The multilayer temporal network of public transport in Great Britain. *Scientific data*, 2:140056, 2015.

[16] R. Gallotti and M. Barthelemy. The multilayer temporal network of public transport in Great Britain., 2015. https://datadryad.org/resource/doi:10.5061/dryad.pc8m3.

[17] Steven Gay, François Fages, Thierry Martinez, Sylvain Soliman, and Christine Solnon. On the subgraph epimorphism problem. *Discrete Applied Mathematics*, 162:214–228, 2014.

[18] W. Han, J. Lee, and J. Lee. Turbo iso: towards ultrafast and robust subgraph isomorphism search in large graph databases. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, pages 337–348. ACM, 2013.

[19] H. He and A. Singh. Graphs-at-a-time: query language and access methods for graph databases. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 405–418. ACM, 2008.

[20] X. He. Interactive template map., 2019. https://hexie1995.github.io/transportation-on-the-map/global.

[21] Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian Active Learning for Classification and Preference Learning. *arXiv:1112.5745 [cs, stat]*, December 2011. arXiv: 1112.5745.

[22] Heinrich Jiang and Maya Gupta. Minimum-Margin Active Learning. *arXiv:1906.00025 [cs, stat]*, May 2019. arXiv: 1906.00025.

[23] A. Jüttner and P. Madarasi. Vf2++—an improved subgraph isomorphism algorithm. *Discrete Applied Mathematics*, 242:69–81, 2018.

[24] Mina Karzand and Robert D. Nowak. MaxiMin Active Learning in Overparameterized Model Classes. *IEEE Journal on Selected Areas in Information Theory*, 1(1):167–177, May 2020. Conference Name: IEEE Journal on Selected Areas in Information Theory.

[25] A. Kopylov and J. Xu. Filtering strategies for inexact subgraph matching on noisy multiplex networks. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 4906–4912, 2019.

[26] L. Kotthoff, C. McCreesh, and C. Solnon. Portfolios of subgraph isomorphism algorithms. In *International Conference on Learning and Intelligent Optimization*, pages 107–122. Springer, 2016.

[27] J. Larrosa and G. Valiente. Constraint satisfaction algorithms for graph pattern matching. *Mathematical Structures in Computer Science*, 12(4):403–422, 2002.

[28] Mauro Maggioni and James M. Murphy. Learning by active nonlinear diffusion. *Foundations of Data Science*, 1(3):271, 2019. Company: Foundations of Data Science Distributor: Foundations of Data Science Institution: Foundations of Data Science Label: Foundations of Data Science Publisher: American Institute of Mathematical Sciences.

[29] M. Magnani and L. Rossi. The ML-Model for Multi-layer Social Networks. In *ASONAM*, pages 5–12. IEEE Computer Society, 2011.

[30] Eric Malmi, Aristides Gionis, and Evimaria Terzi. Active network alignment: a matching-based approach. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1687–1696, 2017.

[31] C. McCreesh and P. Prosser. A parallel, backjumping subgraph isomorphism algorithm using supplemental graphs. In Gilles Pesant, editor, *Int. Conf. on Principles and Practice of Constraint Programming*, pages 295–312. Springer Int. Publishing, 2015.

[32] Ciaran McCreesh, Patrick Prosser, Christine Solnon, and James Trimble. When subgraph isomorphism is really hard, and why this matters for graph databases. *Journal of Artificial Intelligence Research*, 61:723–759, 2018.

[33] Ciaran McCreesh, Patrick Prosser, and James Trimble. The Glasgow subgraph solver: using constraint programming to tackle hard subgraph isomorphism problem variants. In *International Conference on Graph Transformation*, pages 316–324. Springer, 2020.

[34] J. J. McGregor. Relational consistency algorithms and their application in finding subgraph and graph isomorphisms. *Information Sciences*, 19(3):229–250, 1979.

[35] Jacob D Moorman, Qinyi Chen, Thomas K Tu, Zachary M Boyd, and Andrea L Bertozzi. Filtering methods for subgraph matching on multiplex networks. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 3980–3985. IEEE, 2018.

[36] Jacob D Moorman, Thomas Tu, Qinyi Chen, Xie He, and Andrea Bertozzi. Subgraph matching on multiplex networks. *IEEE Transactions on Network Science and Engineering*, 8(2):1367 – 1384, 2021.

[37] Thien Nguyen, Dominic Yang, Yurun Ge, Hao Li, and Andrea L Bertozzi. Applications of structural equivalence to subgraph isomorphism on multichannel multigraphs. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 4913–4920. IEEE, 2019.

[38] J.-C. Régin. A filtering algorithm for constraints of difference in CSPs. In *Proc. of the 12th Nat. Conf. on Artificial Intell, Seattle, WA, USA, July 31 - August 4, 1994, Volume 1.*, pages 362–367, 1994.

[39] M. Puck Rombach, Mason A. Porter, James H. Fowler, and Peter J. Mucha. Core-periphery structure in networks. *SIAM Journal on Applied Mathematics*, 74(1):167–190, 2014.

[40] C. Schwartz. Modeling Adversarial Activity (MAA), 2018. "(2018, Oct 2)".

[41] Francesc Serratosa and Xavier Cortés. Interactive graph-matching using active query strategies. *Pattern Recognition*, 48(4):1364–1373, 2015.

[42] Burr Settles. Active Learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–114, June 2012.

[43] C. Solnon. AllDifferent-based Filtering for Subgraph Isomorphism. *Artificial Intell.*, 174:850–864, August 2010.

[44] Christine Solnon. Experimental evaluation of subgraph isomorphism solvers. In *International Workshop on Graph-Based Representations in Pattern Recognition*, pages 1–13. Springer, 2019.

[45] Christine Solnon, Guillaume Damiand, Colin De La Higuera, and Jean-Christophe Janodet. On the complexity of submap isomorphism and maximum common submap problems. *Pattern Recognition*, 48(2):302–316, 2015.

[46] D. Sussman, Y. Park, C. E. Priebe, and V. Lyzinski. Matched filters for noisy induced subgraph detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2019.

[47] Simon Tong and Daphne Koller. Support Vector Machine Active Learning with Applications to Text Classification. *Journal of Machine Learning Research*, 2(Nov):45–66, 2001.

[48] Thomas Tu. A constraint propagation approach for identifying biological pathways in COVID-19 knowledge graphs, 2021.

[49] Thomas K. Tu, Jacob D. Moorman, Dominic Yang, Qinyi Chen, , and Andrea L. Bertozzi. Inexact attributed subgraph matching. *Proc. IEEE Cong. BIG DATA, Graph Techniques for Adversarial Activity Analytics (GTA3 4.0) workshop*, pages 2575–2582, 2020.

[50] J. R. Ullmann. An algorithm for subgraph isomorphism. *J. ACM*, 23(1):31–42, January 1976.

[51] J. R Ullmann. Bit-vector algorithms for binary constraint satisfaction and subgraph isomorphism. *Journal of Experimental Algorithmics (JEA)*, 15:1–6, 2010.

[52] United Kingdom's national public transport data repository, 2015. https://data.gov.uk/dataset/d1f9e79f-d9db-44d0-b7b1-41c216fe5df6/national-public-transport-data-repository-nptdr.

[53] Dominic Yang, Yurun Ge, Thien Nguyen, Denali Molitor, Jacob Moorman, and Andrea Bertozzi. Equivalence in subgraph matching, 2021.

[54] S. Zampelli, Y. Deville, and C. Solnon. Solving subgraph isomorphism problems with constraint programming. *Constraints*, 15:327–353, 07 2010.

[55] P. Zhao and J. Han. On graph query optimization in large networks. *Proceedings of the VLDB Endowment*, 3(1-2):340–351, 2010.

[56] Xiaojin Zhu, John Lafferty, and Zoubin Ghahramani. Combining Active Learning and Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions. In *ICML 2003 workshop on The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*, pages 58–65, 2003.