Multi-Constrained Graph Pattern Matching in Social Network

Khan Muhammad Atif[1[45697647]]

[1] Department of Computing, Macquarie University, NSW 2109, Australia
lncs@springer.com

Keywords: Online Social Network (OSN), Strong Social Context (SSC), Multi-Constrained Graph Pattern Matching (MC-GPM)

1 Introduction

1.1 Background

A Graph database is a NoSQL database that treats the relationships between data as equally important to the data itself. The data is stored using the graph data model comprised of vertices/nodes, which is an entity such as a person, place, object, or relevant piece of data and edges, which represent the relationship between two nodes. (OrientDB, n.d.) Nodes can hold any number of key-value pairs called properties. Similarly, relationships can also have properties, which are usually quantitative properties like eights, costs, distances, ratings, time intervals, or strengths. (neo4j, n.d.)

Context-aware data mining is related to how the attributes should be interpreted under specific request criteria (Singh, Vajirkar, & Lee, 2003). Context-aware graph data mining is a methodology implemented in domains where context information is represented and stored in the form of a graph.

At present, the fast-paced expansion and advancement in science and technology extended the implementation of graphs in various disciplines like biology, medical science, physics, and chemistry. A method to query data graphs is Graph Pattern Matching *(GPM)* that is originally defined in terms of subgraph isomorphism (exact match), in which, given a data graph G_D and a pattern graph G_P as input, it is to find all subgraphs GM in G_D that are isomorphic to G_P (Fan, et al., 2010). In its simplest form the graph pattern matching refers to a technique that discovers all possible pairs of a pattern graph G_P in a data graph G_D.

Amongst the many applications, GPM is widely implemented in networking data due to the great expressive power for describing and understanding relationships and their complexities. A graph pattern can be specified as to find entities that can be people, organizations, etc. that are in some way linked to each other. And for this reason, *GPM* is significantly used in the compact and congested Online Social Network *(OSN).*

1.2 Motivation

Conventional querying and information extraction from a data graph defined by subgraph isomorphism was too restrictive to find all possible patterns in real-life networks and emerging applications such as *OSNs*.

To cater to the above an improvement popped up in the form of graph simulation and upgraded the conventional *GPM*. Graph simulation, however, is not as rigid as graph isomorphism.

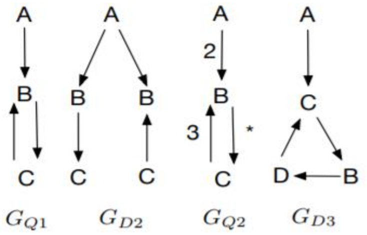

Fig. 1. (Liu, et al., 2015)

G_{Q1} is not isomorphic to G_{D2}, but it matches G_{D2} via graph simulation as *B* and *C* in G_{Q1} can be simulated to one out of *B* and *C* in G_{D2}, respectively. Although the graph simulation was largely adopted, the limitation for applications that expect not just edge-to-edge mapping but also required examining the connectivity of nodes via a path of an arbitrary length or with a bound on the number of hops (Fan, et al., 2010) remained an issue.

The answer to this issue was found in a bounded simulation technique where a match is found in the given data if the bounds condition for nodes is satisfied. This enabled the mapping of the edges in pattern graph to paths in the data graph within bounded lengths. The digits in G_{Q2} links represent max path length (no of intermediary links) between 2 nodes and the * means no restriction of path length. G_{D3} is a match of the pattern graph G_{Q2} that satisfies all the bound constraints.

Bounded simulation served quite well but it only focused on the path lengths. For context-aware data mining like in Contextual Social Graphs *(CSG)*, the existing methods do not consider the social contexts that influence social interactions between participants. For a context-aware network discovery the quality of trust network bears significant importance and its computation requires each vertex to have the social role information, and each edge to have the social relationships and social trust information. (Liu, et al., 2015) proposed a multi constrained graph pattern matching and later improved his work by extending it to distributed computing structure. This "*MCS-GPM: Multi-Constrained Simulation-Based Graph Pattern Matching*" is the technique that will be introduced and discussed in this research report.

2 Related Work

To date the literature contains numerous techniques to solve the Graph Pattern Matching problem. Apparently, all these techniques can be categorized under two main classifications: 1. Sub-graph Isomorphism 2. Graph Simulation. Within a graph-simulation are defined *bounded graph simulation* and *multi-constrained graph simulation*. We will analyze these techniques in detail.

2.1 Sub-Graph Isomorphism

Graph isomorphism as mentioned earlier tries to find the exact match of a pattern graph G_P in the data graph G_D. Graph isomorphism is rigid in its pattern matching and looks for a one-to-one relation between the corresponding pair of nodes and edges in pattern graph G_P and the data graph $G_{D.}$ The extensive search for each pair of nodes classify the detection of all possible subgraphs as an expensive approach in terms of time and adds to the complexity of the technique. Existing work on subgraph isomorphism focuses mainly on precomputing graph structure information and for optimized querying indexing technique is mostly exploited.

R-Join Pattern Matching

(Cheng, Yu, Ding, Yu, & Wang, 2008) proposed processing graph pattern matching as a sequence of *R-join* (reachability-join) that indexes the vertex within two hops away, upon a graph database that stores a data graph in the form of tables.

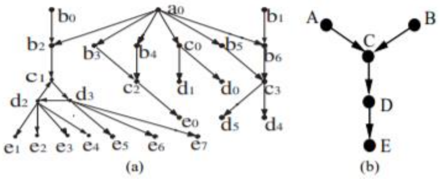

Fig. 2. Data Graph (a) & Graph Pattern (b) (Cheng, Yu, Ding, Yu, & Wang, 2008)

In the approach, they came up with labeling the data graph G_D where, *V is a set of nodes/vertices, E is a set of edges,* , *mapping function that assigns* $v_i \in V$.

Every reachability condition $X \rightarrow Y \in E(G_P)$ is a join i.e. every edge in pattern graph is considered a join. *R-join* is possible based on 2-hop reachability labeling where every node $v \in V(G_D)$ is assigned label $L(v) = (L_{in}(v), L_{out}(v))$ and $L_{in}(v), L_{out}(v) \subseteq V$. The rule for a join between nodes *u and v* is defined in eq. (1).

The 2-hop reachability labeling for G_D is derived from a 2-hop cover of G_D such as $S\ (U_W\ w,\ V_W),\ W \in V(G_D)$ *and is considered as the center node,* and $U_W,\ V_W \subseteq V(G_D)$ *Implies:* $u \rightarrow w,\ w \rightarrow v$ *thus,* $u \rightarrow v$

The scheme proposed the storing of the graph (G_D) in a database G_{DB} with node-oriented representation. The number of tables in G_{DB} each comprising 3 fields

X, X_{in}, X_{out}. R-join needs to check the reachability condition $\ddot{X}\rightarrow Y$ at run time, which incurs high cost. Therefore, a cluster-based R-join index for a data graph G_D using a B^+ *tree* is used.

Distance-Join GPM

Given a large graph G, a query graph Q with n vertices and a parameter δ, n vertices in G match Q iff these *n* vertices in *G* have the same labels as the corresponding vertices in *Q*, and for any two adjacent vertices v_i and v_j in *Q* (i.e., there is an edge between v_i and v_j in *Q* and), the distance between two corresponding vertices in *G* is no larger than (Zou, Chen, Ozsu, & Zhao, 2009) utilized the shortest path to measure the distance between two vertices

Similarity-Based GPM

Given a graph database $D = \{G_1, G_2, . . ., G_N\}$ and a query graph *Q*, similarity search is to discover all the graphs that approximately contain this query graph. The reverse similarity search is to discover all the graphs that are approximately contained by this query graph (Yan, Yu, & Han, 2005).

The similarity measures can be classified into three categories: physical property-based: e.g., toxicity and weight, feature-based: number of common elementary structures determine the similarity, structure-based: directly compares the topology of two graphs

An improvement to these similarity measures was introduced by (Zhu, Lu, Xu, & Hong, 2012) based on the maximum common subgraph *(MCS)* of two graphs. The method worked its way out by breaking a data graph G_D into several groups of similar graphs. For efficient pruning, they derived two distance lower bounds and for further efficiency exploited a triangle property of similarities among the query graph and two database graphs with the help of indexing.

2.2 Graph Simulation

Graph simulation is a technique that revolutionized the traditional subgraph isomorphism and served the detection of more subgraphs in data graph G_D that matches the pattern graph G_P alongside adding to the efficiency of these discovered subgraphs. Graph simulation introduced relaxation to the strict exact match requirement for graph pattern matching. Instead of looking for a one-to-one relation between corresponding nodes and edges, graph simulation allows for a variable count of nodes and edges.

Bounded Simulation

A concept of bounded graph simulation was suggested by (Fan, et al., 2010) where a match for G_P is a relation instead of a function, usually, subgraph isomorphism uses bijective functions. This flourished the idea that for a node *u* in G_P there exists a non-empty set of nodes *v* in G_D and each pair *(u, v)* satisfies the relation. Bounded simulation extends the conventional edge-to-edge mapping in graph simulation and allowed for edges to be mapped on paths comprising several bounds rather than just edge to edge mappings.

Multi-Constrained Simulation

Multi-Constrained Simulation (*MCS*) is an improved version of the bounded simulation. In addition to the bounded path lengths, the pattern graph G_P contains multiple constraints on the edges. These additional constraints enabled the Multi-Constrained Graph Pattern Matching (*MC-GPM)* built upon *MCS,* to extract out the subgraphs that have attributes defined on vertices and edges.

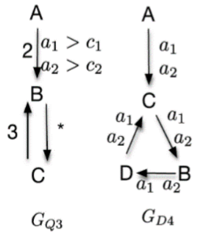

Fig. 3. (Liu, et al., 2015)

The pattern graph G_{Q3} defines additional constraints of some more attributes other than bounds associated with the edge *A→B*. (Liu, et al., 2015)'s *MCGPM* method looks for a match in G_{D4} for *(A, B)* such that the path length does not exceed the maximum bound 2 and also check for the aggregated values of *a and b* to satisfy the defined constraints of $a_1 > c_1$ *and* $a_2 > c_2$. Different requirements may have varying constraints (e.g., total cost<$100, delay<10s, and availability>60%).

3 Identified Methodologies

3.1 Problem Statement

The core issue is to find all matches in an arbitrary large directed data graph G_D that match all the reachability conditions conjunctively specified in a graph pattern, G_Q. The high time cost and exhaustive checking for each pair of nodes rendered subgraph isomorphism, as an NP-Complete problem.

Although bounded simulation brought improvements to finding the closest matched subgraph it cannot support more than a single constraint, i.e., is the bounded path length. In the real-world scenario, there are diverse applications in social networks like crowd-sourcing travel (Milano, Baggio, & Piattelli, 2011), study group selection, social network-based e-commerce, etc. there are multiple social contexts that are critical for collaborations and decision making.

According to (Kuter & Golbeck, 2007) trust is one of the most important factors for participants' decision-making in online social networks. In an environment where participants are physically unknown to each other and it is challenging to establish the trustworthiness of a participant. Based on the importance of trust and *Quality of Service (QoS)*, (Guanfeng, Yan, & A., 2010) defined *Quality of Trust (QoT)* as a degree of trustworthiness in a path with constraints like 1)Trust aggregation (*T*), 2)Social intimacy degree aggregation (*r*), 3)Role impact factor ( ) and marked them vital for *GPM.*

The sub-graph isomorphism and bounded simulation methods do not leverage social intimacy and trust constraints (Liu, et al., 2015). They lack any means to define more than one constraint on the edges in graphs like Contextual Social Graph (*CSG*). Moreover, the existing compression methods require the *GPMs* to decompress the original graph from compact structures to answer the graph pattern query, thus adding to the complexities, computation, and time cost.

Example:

Consider figure4. Let A be a professor in a university and let B be one of the research assistants working under the professor. If the requirement is to find a dedicated and hardworking research team there will be a need to determine additional constraints. Let $a_1 > c_1$ *and* $a_2 > c_2$ be the validation rule for trust and social intimacy constraints. The existing methods of *GPM* do not support these multiple constraint queries.

Existing *GPM* graph compressions are restricted decompress the original graph from compact structures to answer the graph pattern query. A solution viable to utilize distributed computing to extract pattern graphs as output without having to decompress the data graph when supplied with social contexts and a query graph with the constraints of the bounded path length and social contexts as inputs.

3.2 The proposed method

After establishing the importance of a way to deal with multiple constraints in *GPM* this section explains the *MC-GPM* in a context-aware *Online Social Network (OSN)*.

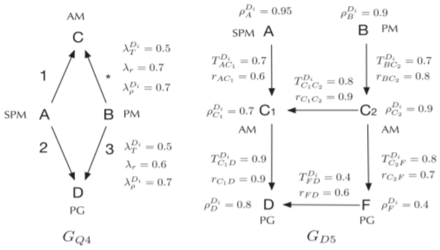

Fig. 4. Multiple-Constrained GPM in CSGs (Liu, et al., 2018)

A data graph G_{D5} is labeled G_D = *(V, E, LV, LE)*, where, *V is a set of vertices (A, B,* C_1, C_2, *D, F), E is a set of edges (*$A \rightarrow C_1$, $B \rightarrow C_2$ *, LV is a function that maps node labels, and LE is a function that maps edge (relationship) labels*.

A pattern graph G_{Q5} is labeled G_q = *(*V_q, E_q, f_v, f_e, s_e*)*, where, V_q *is a set of vertices,* E_q *is a set of edges,* f_v *is a function such that* f_v *(u) is the vertex label of u,* f_e *is a func-*

tion such that $f_e(u, u`)$ *is the bounded length of* $(u, u`)$ *e.g. bounds (1, 2, 3,*), and* $s_{e,}$ *is a function such that* s_{e} $(u, u`)$ *is the multiple constraints of aggregated social impact* $_T$ $_r$ *=0.7)*

Each vertex $v_i \in V$ is associated with the role impact factor ($^{Di}{}_{vi}$) $\in [0, 1]$, representing the expertise of v_i in domain *i*. Every edge (v_i, v_j) has $T^{Di}{}_{vi,\, vj} \in [0, 1]$, and $r_{vi,\, vj} \in [0, 1]$ to denote social trust in domain *i*, and social intimacy between vertices. Figure 5 illustrates bounds and additional constraints in pattern graph G_Q and data graph G_D.

The *MCS* baseline algorithm finds a match in G_D for G_Q by the existence of a binary relation $S \subseteq V_Q \times V$. The algorithm looks for a path that first satisfies the bounds defined in G_Q, such that for each edge $(u, u`)$ in $E_Q$ there exists a path *p from v to v`* in G_D that confirms $(u`, v`) \in S$ and $Slen(p) \leq k$, *provided* $f_e(u, u`) = k$. The solution is then followed by the probe if the detected path fulfills the multiple constraints validation criteria of the social context provided in pattern graph G_Q, such that AT^{Di} $_T$ $_r$ ^{Di} *, given that* S_e $_T$ $_r$ *).* The algorithm then repeats itself to find the best solution.

To address the efficiency and effectiveness of the *MC-GPM* the method looks for *Strong Social Components (SSC)* in G_D. *SSC* is defined as a subgraph that in a specific domain contains a high role impact factor value for each of its vertices, so the *MC-GPM* randomly selects *k* such vertices and adopts Breadth-First Search (BFS) method for the later portion of the definition that explains about the edges being marked with strong social trust and social intimacy relationships (Liu, et al., 2018).

Aware of the fact that stable structure does not change very frequently, and they can be used this for compressions of structure with a low update cost. Therefore, the social context preserved[1] compression method preserves not only the reachability and pattern but also saves the social context. This way the compressed form contains all useful information and there remains no further need for the decompression for pattern matching.

To further enhance the efficiency and optimize the algorithm, *MC-GPM* indexes the reachability, graph pattern, and social contexts in compressed *SSCs.* The significantly small size of *SSC* compared to the entire data graph reduces the computation cost for indexing, particularly the computation of the reachability index. The 3-tier indexing structure stores at each vertex the reachability information by identifying ancestor and descendant nodes, the pattern graph information by recording the shortest path length *Slen* between any two vertices in the *SSC*, and finally the social context by recording the maximal aggregated social impact factor values of the mapped paths in a data graph.

Example

[1] The compressed graph is social context preserved when two compressed vertices have the same label, the same ancestors, the same descendants, and the aggregated social impact factor values of the path via one of the vertex dominates that of the other one. (Liu, et al., 2018)

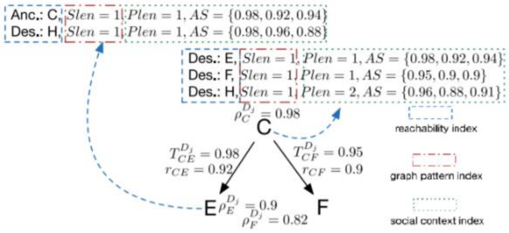
Anc.: C, $Slen = 1$, $Plen = 1, AS = \{0.98, 0.92, 0.94\}$
Des.: H, $Slen = 1$, $Plen = 1, AS = \{0.98, 0.96, 0.88\}$
Des.: E, $Slen = 1$, $Plen = 1, AS = \{0.98, 0.92, 0.94\}$
Des.: F, $Slen = 1$, $Plen = 1, AS = \{0.95, 0.9, 0.9\}$
Des.: H, $Slen = 1$, $Plen = 2, AS = \{0.96, 0.88, 0.91\}$
$\rho_C^{D_j} = 0.98$
C
$T_{CE}^{D_j} = 0.98$
$r_{CE} = 0.92$
$T_{CF}^{D_j} = 0.95$
$r_{CF} = 0.9$
E
$\rho_E^{D_j} = 0.9$
$\rho_F^{D_j} = 0.82$
F
reachability index
graph pattern index
social context index

Fig. 5. The index of an SSC in domain j (Liu, et al., 2018)

The method strives for improved efficiency and adopts a multi-threaded approach to parallel processing the NP-Complete MC-GPM and bidirectionally looks for a feasible edge pattern match *F-EPM*. For the final output, it searches for the shortest path in parallel to discovers in the Optimal edge pattern matching *(O-EPM)* the edge matching with the minimal bounded path length in the data graph.

4 Conclusion and Future Work

The proposed *MC-GPM* is an example of efficient utilization of existing methodologies and an effective expansion from a simple single-constrained graph simulation. The work deploys state-of-the-art techniques in pathfinding, graph compression, and graph indexing. Multi-constrained path problem is NP-Complete and therefore the method proposes heuristic matching strategies that bidirectionally finds Edge Pattern Matching in parallel for both F-EPM and O-EPM processes, which improves the efficiency of graph search. The method not only inspires itself from technology but base the durability of the connection structure on findings of social psychology theory upon which the recommended indexing and compression competently save the processing time and the search space.

Although, the social-context serves the pivot for *MC-GPM*, but it has every possibility to be extended to other disciplines that could benefit from productive and optimal graph search. Very importantly for example, in the current pandemic we can monitor use the social graphs to track for infected people and can look for their relations with others, identify who the meet and this way we could possibly trace down the infected individuals and take necessary actions to prevent further spread.

As there is always room for improvement the *MC-GPM* could possibly optimize the future searches by caching the results of queried graphs as patterns for later reference. Instead of carrying out a detailed search every time answers can be found in the cache. For the new queries if similar solutions, extend from the existing cached references, then the new results can use the cache graphs as the starting point for their search. These cached referencing could possibly reduce the computation and processing time at the cost of some additional storage.

References

Cheng, J., Yu, J. X., Ding, B., Yu, P. S., & Wang, H. e. (2008). Fast Graph Pattern Matching. *2008 IEEE 24th International Conference on Data Engineering* (pp. 913-922). Cancun: IEEE.

Fan, W., Li, J., Ma, S., Tang, N., Wu, Y., Wu, Y., & en. (2010). Graph Pattern Matching: From Intractable to Polynomial Time. *Proceedings VLDB Endowment*, 264-275.

Guanfeng, L., Yan, W., & A., O. M. (2010). Optimal Social Trust Path Selection in Complex Social Networks. *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010*, (pp. 1391–1398). Atlanta.

Kuter, U., & Golbeck, J. (2007). SUNNY: A New Algorithm for Trust Inference in Social Networks Using Probabilistic Confidence Models. *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence.* Vancouver.

Liu, G., Liu, Y., Zheng, K., Liu, A., Li, Z., & Yang Wang, e. a. (2018). MCS-GPM: Multi-Constrained Simulation Based Graph Pattern Matching in Contextual Social Graphs. *IEEE Transactions on Knowledge and Data Engineering, vol. 30, no. 6*, 1050-1064.

Liu, G., Zheng, K., Wang, Y., Orgun, M. A., Liu, A., Zhao, L., & Zhou, X. (2015). Multi-Constrained Graph Pattern Matching in large-scale contextual social graphs. *2015 IEEE 31st International Conference on Data Engineering* (pp. 351-362). Seoul: IEEE.

Milano, R., Baggio, R., & Piattelli, R. (2011). The effects of online social media on tourism websites. *18th International Conference on Information Technology and Travel & Tourism.* Innsbruck: Springer, Vienna.

neo4j. (n.d.). *What is a Graph Database*. Retrieved from neo4j: https://neo4j.com/developer/graph-database/

Nguyen, T. V., Lim, W., Nguyen, H. A., & Choi, D. (2008). Context Awareness Framework based on Contextual Graph. *IEEE Xplore*, 1 - 5.

OrientDB. (n.d.). *Graph database*. Retrieved from OrientDB: https://orientdb.com/graph-database/

Singh, S., Vajirkar, P., & Lee, Y. (2003). Context-aware Data Mining using Ontologies.

Yan, X., Yu, P. S., & Han, J. (2005). Substructure Similarity Search in Graph Databases. *in Proc. ACM SIGMOD Int. Conf. Manag. Data*, (pp. 766–777).

Zhu, Y., Lu, Q., Xu, Y. J., & Hong, C. (2012). Finding Top-K Similar Graphs in Graph Databases. *Proceedings of the 15th International Conference on Extending Database Technology*, (pp. 456-467).

Zou, L., Chen, L., Ozsu, M. T., & Zhao, D. (2009). DistanceJoin: Pattern Match Query In a Large Graph Database. *in Proc. 30th Int. Conf. Very Large Data Bases*, (pp. 886-897).