Fast Best-Effort Search on Graphs with Multiple Attributes

(Extended Abstract)

Senjuti Basu Roy
Institute of Technology
University of Washington Tacoma
Email: senjutib@uw.edu

Tina Eliassi-Rad
Department of Computer Science
Rutgers University
Email: eliassi@cs.rutgers.edu

Spiros Papadimitriou
Business School
Rutgers University
Email: spapadim@business.rutgers.edu

***Abstract*—We address the problem of top-k search on graphs with multiple nodal attributes, which we call *WAGs* (short for *Weighted Attribute Graphs*). For example, a co-authorship network is a WAG, where each author is a node; each attribute corresponds to a particular topic (e.g., databases, data mining, and machine learning); and the amount of expertise in a particular topic is represented by a non-negative weight on that attribute. A typical search in this setting may be: find three coauthors (i.e., a triangle) where each author's expertise is greater than 50% in at least one topic area (i.e., attribute). We show that the problem of retrieving the optimal answer for graph search on WAGs is NP-complete. Moreover, we propose a fast and effective top-k graph search algorithm for WAGs. In an extensive experimental study, our proposed algorithm exhibits significant speed-up over competing approaches. On average, our proposed method achieves 7× faster query processing than the best competitor.**

I. Introduction

We are interested in top-k search problems on *WAGs* (short for *Weighted Attribute Graphs*), where (1) nodes have multiple attributes (e.g., an author has multiple areas of expertise) and (2) attributes on nodes have non-negative weights associated with them (e.g., an author has varying degrees of expertise across different topics). Examples of WAGs include co-authorship networks with multiple expertise as attributes, friendship networks with various user activities (such as liking, commenting, or clicking on different types of posts) as attributes, (3) communication networks with various modalities used by each person (such as % instant messages, % emails, % phone calls) as attributes, *etc.*

An important task on WAGs is to quickly and effectively answer a user's search, where the search can include (1) constraints on the connectivity of nodes and (2) constraints on the weighted attribute values. In this work, we present a fast, best-effort solution for top-k search on WAGs. We normalize the weights on attributes per-node (i.e., the sum of the weights over all attributes for each node is one). While this normalization makes the query semantics cleaner, our approach can also be easily adapted to other types of normalization. Figure 2 depicts an example WAG and search query.

II. Background & Problem Definition

A graph query is defined by $H_q = (V', E')$ and and W_q; the former is the graph query's structure, while the latter is a node × attribute matrix with attribute-weights for each node. Relative to the data graph, H_q is a small graph. Based on two different types of queries on the nodes of H_q, we further define *point graph query* and *range graph query*.

Point queries require each $w_{i,j}$ to be a single (i.e., point) value between 0 and 1 with rows of W_q summing to 1, whereas range queries permit greater flexibility by allowing (i) omission of weights for some of the attributes and some of the nodes; and (ii) specification of weight ranges, rather than point values.

Ranking Function (R-WAG)**.** When ranking the results of a query, we want the ranking function to consider both the attribute and the structural divergences. The Jensen Difference is able to represent both of these differences; thus, we use it as `R-WAG`'s ranking function. Suppose G_s is a candidate subgraph in response to the graph query H_q, then our ranking of G_s with respect to H_q is defined as follows:

$$F(G_s, H_q) = \sum_{i=1}^{|V_s|} D(v_i, v_i')$$

where $v_i \in G_s$ and $v_i' \in H_q$. Then, the divergence function $D(v_i, v_i') = 1$ if v_i is an unmatched node. Otherwise, $D(v_i, v_i') = \mathit{JensenDiff}(v_i, v_i')$. Figure 1 shows an example of our divergence score calculation.

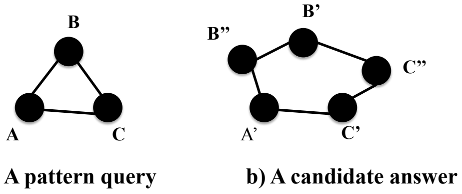
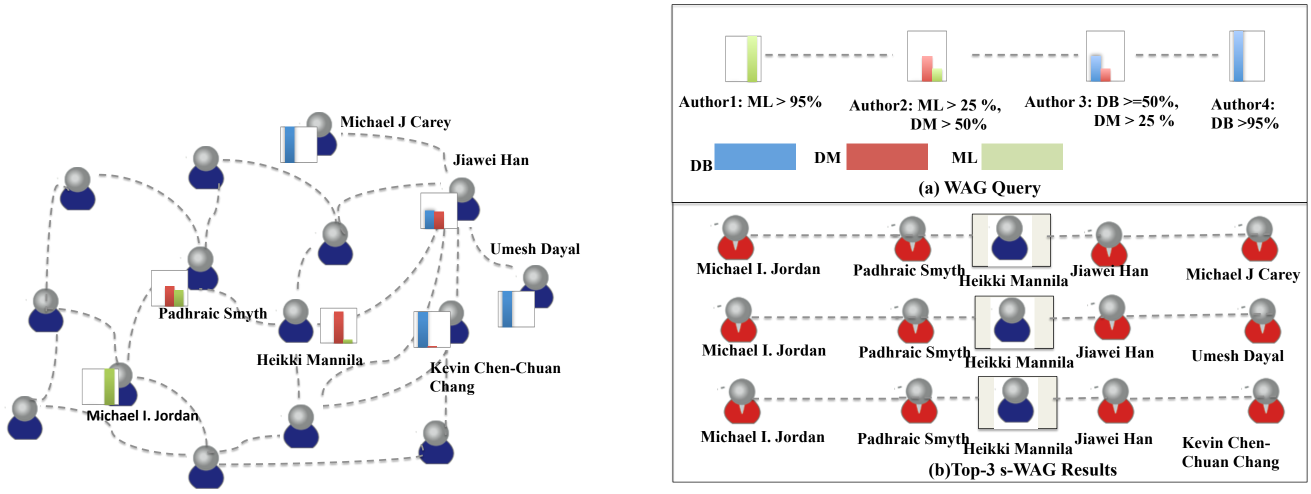


Fig. 1. An example pattern query in (a) and a candidate answer in (b). The ranking score of the candidate subgraph is: $\mathit{JensenDiff}(A, A') + \mathit{JensenDiff}(B, B') + \mathit{JensenDiff}(C, C') + 1 + 1$. A Jensen Difference of 1 is added for each of the unmatched nodes (namely, B$''$ and C$''$) to warrant approximate matching over the structure.

Problem Definition: Top-k Graph Search on a WAG. Given a WAG, a graph query (point or range), and an integer k, identify a set of k subgraphs of the WAG such that (1) the k subgraphs are ranked in ascending order of their overall divergence scores (F) with respect to the graph query; and (2) any subgraph that is not present in the set has a larger divergence score with respect to the graph query compared to the overall divergence score of the k-th subgraph.



(a) A partially constructed DBLP co-authorship network

(b) A query over the DBLP co-authorship network and top-3 results returned by s-WAG

Fig. 2. (a) Each node in this WAG (from the DBLP co-authorship network) is an author with three attributes: expertise in databases (DB), data mining (DM), and machine learning (ML). A graph query may wish to find a path of 4 authors (structural constraint) such that it connects ML researchers to DB researchers using DM researchers as intermediaries. The latter property is enforced by using weights on nodal attributes (weight constraints; depicted by "bar heights"). Such a search with path structure and range constraints on attribute-weights is depicted on the top-half of (b). The bottom half of (b) describes the 3-best answers returned by our s-WAG. Author Heikki Mannila acts as "bridge" (primary expertise DM) to connect authors who are somewhat uniformly spread between ML, DM (Padhraic Smyth) and DB, DM (Jiawei Han).

III. Summary of Technical Contributions

The major contributions of this work are as follows. **(1)** We show that the top-k WAG search problem is NP-complete, by reducing an instance of sub-graph isomorphism to an instance of our problem. **(2)** When addressing the problem of search on a WAG, we consider issues such as structure vs. weighted-attribute matching, point vs. range queries on weighted-attributes, exact vs. inexact algorithms, and optimal vs. approximate solutions. Our approach addresses *all* these issues with three components: I-WAG, s-WAG, and R-WAG. Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and a matrix $\mathcal{W}$ whose w_{ij} entry corresponds to the weight of attribute j on node i, I-WAG builds a *hybrid* index on the graph that incorporates both the weighted attributes and the structure of the network. Upon receiving a search query, s-WAG utilizes the output of I-WAG to quickly retrieve the best k matches based on both weighted attributes and structure. R-WAG ranks the results by utilizing a novel ranking function that unifies the structural and nodal divergence and is monotonic in nature. **(3)** We show that the top-k results returned by s-WAG are optimal in the context of answer quality, considering the ranking function in R-WAG (described in Section II).

IV. Experimental Evaluation

We perform both qualitative and quantitative experiments on several real-world datasets with millions of nodes and edges and various node-attribute characteristics. See [2] for details.

Baseline Algorithms: We appropriately modify an existing work, called G-ray [3], that efficiently supports search on graphs with unweighted, single attributes (i.e., each vertex has a single attribute and no weights). We primarily report comparative studies between s-WAG and this modified competitor, which we call WAG-ray. The performance of other baseline methods is typically worse, as detailed in [2].

Queries: We primarily consider four different types of structures for graph query—namely, star, path, loop, and clique. For our performance experiments, a point query vertex is generated from the underlying $\mathcal{W}$ matrix uniformly at random. A range query vertex is generated from a point query, where we arbitrarily introduce ranges on the specific attribute weights. In either case, the graph search is one of the four aforementioned graph queries.

Summary of Results: **(1)** We present the results of an extensive case study on YouTube data, where the attributes are produced by a mixed-membership role-discovery algorithm [1]. In this study, we pose multiple graph search queries. Our results show that graph search on WAGs is a powerful data exploration tool and that s-WAG is an effective solution towards that end. **(2)** Build times for both WAG-ray and s-WAG scale linearly with increasing graph size. At the same time, WAG-ray almost consistently outperforms s-WAG with respect to build time. This is unsurprising, since the indices in WAG-ray are rather "shallow," as it was not originally designed to perform search over WAGs. The slightly higher build time in s-WAG is compensated by its very efficient query processing time. **(3)** s-WAG outperforms WAG-ray consistently in query processing time, irrespective of the sparsity of the weighted attribute matrix. Overall, considering both build time and query processing time, our results demonstrate that the combination of I-WAG and s-WAG is a better graph search approach than WAG-ray for WAGs, exhibiting up to 7× better query response times. For details, please see [2].

References

[1] K. Henderson, B. Gallagher *et al.*, "RolX: Structural role extraction & mining in large graphs," in *KDD*, 2012, pp. 1231–1239.

[2] S. B. Roy, T. Eliassi-Rad, and S. Papadimitriou, "Fast best-effort search on graphs with multiple attributes," *IEEE TKDE*, vol. 27, no. 3, pp. 755–768, 2015.

[3] H. Tong, C. Faloutsos, B. Gallagher, and T. Eliassi-Rad, "Fast best-effort pattern matching in large attributed graphs," in *KDD*, 2007, pp. 737–746.