Mining Social Ties Beyond Homophily

Hongwei Liang
School of Computing Science
Simon Fraser University, Canada
hongweil@sfu.ca

Ke Wang
School of Computing Science
Simon Fraser University, Canada
wangk@cs.sfu.ca

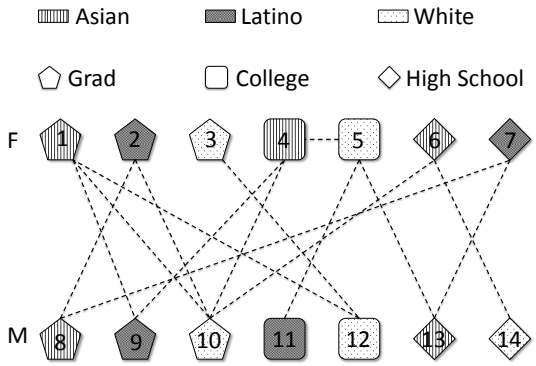
Feida Zhu
School of Information Systems
Singapore Management University, Singapore
feida@smu.edu

***Abstract*—Summarizing patterns of connections or social ties in a social network, in terms of attributes information on nodes and edges, holds a key to the understanding of how the actors interact and form relationships. We formalize this problem as *mining top-k group relationships (GRs)*, which captures strong social ties between groups of actors. While existing works focus on patterns that follow from the well known homophily principle, we are interested in social ties that do not follow from homophily, thus, provide new insights. Finding top-k GRs faces new challenges: it requires a novel ranking metric because traditional metrics favor patterns that are expected from the homophily principle; it requires an innovative search strategy since there is no obvious anti-monotonicity for such GRs; it requires a novel data structure to avoid data explosion caused by multidimensional nodes and edges and many-to-many relationships in a social network. We address these issues through presenting an efficient algorithm, GRMiner, for mining top-k GRs and we evaluate its effectiveness and efficiency using real data.**

I. Introduction

Information networks, such as social networks, citation networks, dating networks, etc., are heterogeneous and multidimensional [1] in that nodes and edges belong to certain classes and each class has description on multiple attributes. For example, in addition to links, Facebook contains large quantities of user demographic data that reveal detailed personal information. The frequent patterns of connections in social networks, concisely in terms of attribute information of nodes and edges, indicate specific common social interactions. We call this kind of patterns "social ties". Summarizing such social ties holds a key to the understanding of how the actors interact with each other and form relationships, which is useful in user behavior analysis and modelling, friends/items recommendation, inferring user demographics, etc.

In addition, it is well known that social ties follow the *homophily principle*, or "love of the same" [2]: a contact between similar people occurs at a higher rate than among dissimilar people, where similarity is measured by common characteristics such as beliefs/religion, value, race, age, etc. So far, the literature largely focuses on applications based on the homophily principle, such as community detection, link prediction, friend/product recommendation and information diffusion. However, there is more and more voice from the academic circle and industry that they want to break the boundaries to unearth the treasures beyond homophily. For example, the authors in [3] claim that "recommending popular items is unlikely to result in more gain than discovering insignificant yet liked items because the popular ones might be already known to the user", and [4] infers networks of product relationships to recommend complementary products in addition to substitutes (similar products). Similarly, the social ties following the homophily principle are usually well-expected and people can easily dope out them without much effort, even though they are somewhat useful. Therefore, discovering the social ties that are popular and interesting, but are not simply expected from homophily is more practical. In this paper we study this problem.



(a) Network topology (patterns represent Race and shapes represent Education)

ID	SEX	RACE	EDU
1	F	Asian	Grad
2	F	Latino	Grad
3	F	White	Grad
4	F	Asian	College
5	F	White	College
6	F	Asian	High School
7	F	Latino	High School
8	M	Asian	Grad
9	M	Latino	Grad
10	M	White	Grad
11	M	Latino	College
12	M	White	College
13	M	Asian	High School
14	M	White	High School

(b) Attributes on nodes

Fig. 1: A toy dating network

A. Motivating Examples

To motivate our work, we consider the toy online dating network in Fig. 1a with the node information in Fig. 1b. A

pair of individuals in a dyadic tie is a dating relationship and each individual has attributes SEX, RACE, and EDU. We can represent a group of links between two groups of individuals by a *group relationship* or *GR*, denoted $l \xrightarrow{w} r$. l and r are the attributes information describing the two groups of nodes and w is the attributes information describing the edges between the groups. GRs serve as the representation of social ties.

Example 1: According to the recent study on Facebook dating app, Are You Interested[1], men tended to prefer Asian women for dating. This finding can be represented by GR1 in the following table.

GR1	(SEX:*M*) → (SEX:*F*, RACE:*Asian*) *supp* = 7/15; *conf* = 7/14
GR2	(SEX:*M*, RACE:*Asian*) → (SEX:*F*, RACE:*Asian*) *supp* = 0; *conf* = 0

The edge descriptor $w = dates$ is omitted. $supp = 7/15$ and *conf* = $7/14$ are two intuitive metrics, support and confidence, originally used for association rule mining [5]. $supp = 7/15$ means that 7 out of the 15 links are involved in this relationship, and *conf* $= 7/14$ means that 7 out of the 14 links originating from the nodes for male go to the nodes for Asian women. Having found GR1, people wonder whether Asian men particularly prefer Asian women, so propose the variation GR2, whose $supp$ and *conf* can be obtained by queries on the data. GR2 and GR1 together suggest that while most men preferred Asian women, Asian men are an exception, which is a finding in Are You Interested. This finding could be interesting to a dating service provider. □

GRs that are expected from the homophily principle usually tend to have a high confidence, and such GRs generally have the form $l \xrightarrow{w} r$ where the values in r occur in l. In this paper, we assume that the homophily principle is known, and our goal is to find interesting GRs not expected from homophily. Example 2 illustrates that such GRs can be potentially useful but they are not ranked high by confidence.

Example 2: Consider the two GRs, GR3 and GR4, listed in the table below.

GR3	(SEX:*F*, EDU:*Grad*) → (SEX:*M*, EDU:*Grad*) *supp* = 4/15; *conf* = 4/6
GR4	(SEX:*F*, EDU:*Grad*) → (SEX:*M*, EDU:*College*) *supp* = 2/15; *conf* = 2/6

Assume that the attribute EDU follows the homophily principle. Therefore, GR3 likely has a high confidence but is not interesting because it is expected from the homophily principle. GR4 likely has a low confidence since GR3 has a high confidence. $supp$ and *conf* are obtained from the data in Fig. 1. A closer inspection of the data reveals that if a female with Grad education does **NOT** want her partner to have Grad education, there is a high chance that she prefers a partner with College education. This preference of *College* education, which is conditioned on the educations other than *Grad*, could be interesting to the dating service provider. Though GR4 correctly captures this relationship, it will not be ranked high by the confidence metric. □

[1]http://huffingtonpost.com/jenny-davis/race-online-dating_b_4449946.html

GR4 is ranked low by the confidence metric because most females with *Grad* education dated male partners with *Grad* education according to the homophily principle. However, if we exclude this "homophily effect" by restricting to the male partners not having *Grad* education, GR4 holds 100% of the time, which indicates a strong preference beyond the homophily principle. This observation motivates a new ranking metric, called **non-homophily preference**. Intuitively, non-homophily preference captures "secondary bonds" beyond the "primary bonds" of homophily. One contributing factor of secondary bonds is *heterophily* [6], i.e., the tendency of individuals to collect in diverse groups. It was shown in that heterophilious networks are better to promote and spread innovations [6]. Thereby, though the primary bond is important in multiple applications, exploring the secondary bond can result in more interesting findings and bring extra value to many businesses. The next example further explains this point.

Example 3: To leverage social influence for promoting products, an obvious strategy for a financial institution is to use GRs following from homophily, such as

$$(\text{JOB} : Lawyer, \text{PRODUCT} : Stocks) \rightarrow (\text{PRODUCT} : Stocks)$$

to promote *Stocks* to the friends, f, of existing customers who are lawyers and have bought *Stocks* (on LHS). This effort fails if most such friends f already bought or do not like *Stocks*. On the other hand, suppose

$$(\text{JOB} : Lawyer, \text{PRODUCT} : Stocks) \rightarrow (\text{PRODUCT} : Bonds)$$

has a high non-homophily preference, that is, among the friends f who do not buy *Stocks*, many buy *Bonds*. This GR can be used to promote *Bonds* to a friend if he/she has not bought *Bonds*, and the high non-homophily preference implies a high adoption rate. □

Indeed, many companies have both e-commerce services and social network services, enabling them to create information networks to mine GRs for economic benefits. For example, Alibaba Group[2] provides various sales services, and has the instant messenger Aliwangwang that builds the social network among customers and vendors. As another example, Facebook Platform[3] allows a third party business to build application based on their platforms. This tool enables integrating the social graph with the customer information owned by the third party business, and applications on facebook.com are allowed to access the graph.

B. Contributions

In summary, we make the following contributions.

- We propose a novel ranking metric called non-homophily preference (Section III-B) to identify strong social ties beyond the homophily principle; we define the problem of mining top-k GRs (Section III-C) to extract k most interesting social ties under the non-homophily preference metric.
- The search space of top-k GRs is large due to multidimensional nodes and edges and the lack of usual anti-monotonicity of non-homophily preference.

[2]http://en.wikipedia.org/wiki/Alibaba_Group
[3]http://en.wikipedia.org/wiki/Facebook_Platform

We first propose a compact data model to store the multidimensional nodes and edges information in social networks, then we present a novel search strategy to enable a new form of anti-monotonicity for non-homophily preference (Section IV). This strategy ensures that only non-trivial GRs that meet a minimum requirement on support and non-homophily preference are enumerated.

- We present an efficient top-k GRs mining algorithm, GRMiner, based on the new data model and the above search strategy (Section V).
- We evaluate our approach on two real world data sets (Section VI), and provide potential extensions of our framework (Section VII).

II. Related Work

Graph mining. Most previous works on graph mining summarizes a large graph by simple statistics, such as degree distributions, hop-plots, clustering coefficients and number of triangles. See surveys [7], [8]. Other works summarize a large graph by densely connected subgraphs [9], [10], network motifs [11], and frequent sub-structures [12], [13]. Link prediction [14] uses neighbourhood information to predict the existence of a link between two nodes. Majority of these works exploit only the topological structure of graphs. The work on community detection with node attributes, such as [15], develops a probabilistic model to model the interaction between network structures and node attributes for detecting overlapping communities. The works like [16] and [17] jointly model network structure and vertex attributes with probability models. The motivation of these works is quite different from ours. [18] focuses on class propagation in a social network using a given influence matrix. Our GRs can serve as the assumed influence matrix. In fact, GRs capture a more general type of influences between sub-populations summarized by RHS and LHS, which makes more sense because strong influences typically exist at sub-population level.

Information network summarization. This body of works considers information networks where nodes and edges have attributes like ours [1], [19], [20], [21]. These works focused on summarizing the *entire* graph, whereas our focus is on identifying strong relationships that exist for certain groups of nodes and certain types of edges. The work in [22] focuses on community detection whereas we focus on strong non-homophily patterns between individuals. They considered multigraphs that allow only values on edges (called dimensions) but not on nodes such as SEX and RACE like ours. Such multigraphs cannot model our graphs with values on both edges and nodes, such as "a male dates a female Asian". The itemsets in [22] can construct rules about an individual like "if X publishes in VLDB, X also publishes in SIGMOD" where VLDB and SIGMOD are dimensions on edges, but cannot construct GRs that aim at a pair (X,Y) of individuals such as "if X is a male, X tends to date a female Asian Y", which is useful to gauge the influence between two persons.

Association rule mining. Support and confidence were first introduced for association rule mining [5] from transaction data. Mining frequent combinations of attribute-values in a relational table was studied as iceberg cube queries [23]. [24] proposed "self-sufficiency" to measure the interestingness of itemsets. Multi-relational data mining [25] generalizes frequent patterns by allowing multiple predicates and variables in a pattern, but it does not consider the issues associated with social networks. As pointed out in Section I, the homophily of social networks requires reconsideration of interestingness metrics and new strategies of pruning. [26] studied mining unexpected rules based on prior knowledge where unexpectedness is measured by similarity between fuzzy terms. Such non-statistical rules cannot be used for social network applications that motivate our work. [26] uses support-based pruning, which is too weak to unearth non-homophily GRs buried deeply in many homophily based GRs. Our interestingness metric is a statistical measure and captures the notion of conditional probability, which is good for inference.

Social structure of social networks. Our work has similarity with the study on social structure of Facebook networks in [27], which focuses on calculating the propensity for two nodes with the same categorical value to form a tie. While their work can be used to quantify and specify homophily attributes in our problem, our focus is on searching for unexpected ties that do not follow from homophily.

III. Problem Statements

A social network is a pair $G = (V, E)$ with V being a set of nodes/vertices and E being a set of directed edges/links. $|V|$ denotes the number of nodes in V and $|E|$ denotes the number of edges in E. Each node in V has descriptions over a fixed set of node attributes and each edge in E has descriptions over a fixed set of edge attributes. Each attribute A has a discrete domain $\{0, 1, \cdots, |A|\}$, where $|A|$ is the domain size, with 0 representing the null value. We consider directed edges; an undirected edge can be represented by a pair of directed edges in the opposite directions.

A subset of nodes of V that share same values a on some node attributes A can be represented by a set of pairs $(A : a)$ called a node descriptor. For example, (SEX:*F*, JOB:*IT*) represents all the nodes having the values (SEX:*F*, JOB:*IT*). Similarly, a subset of edges in E that share same values w on some edge attributes W can be represented by a set of pairs $(W : w)$ called an edge descriptor. Table I summarizes the main notations used in the paper.

TABLE I: Frequently used notations

Notation	Definition
$G(V, E)$	graph with the nodes V and the edges E
l, r, w	three parts of attributes values in a GR $l \xrightarrow{w} r$
$\mathcal{L}$ $\mathcal{R}$ $\mathcal{W}$	attributes for the values in l, r, and w
A^l,A^r	A^l is an attribute in $\mathcal{L}$, and A^r is A^l in $\mathcal{R}$
$l \xrightarrow{w} l[\beta]$	homophily effect, see Eqn. (5)

A. Group Relationships

Definition 1: [GR] A *group relationship (GR)* has the form $l \xrightarrow{w} r$, where l and r are node descriptors and w is an edge descriptor. l is called LHS. r is called RHS. $\mathcal{L}$, $\mathcal{W}$, $\mathcal{R}$ denote the attribute sets for l, w, and r, respectively. □

For GR3 in Example 2, l = (SEX:*F*, EDU:*Grad*), w = (TYPE:*dates*), and r = (SEX:*M*, EDU:*Grad*). This GR says that females with *Grad* education tend to prefer male partners with *Grad* education. The "tendency" can be measured by support and confidence [5].

Definition 2: [Support] *Support* of $l \xrightarrow{w} r$ indicates the probability that an edge satisfies all the conditions in $l \wedge w \wedge r$:

$$supp(l \xrightarrow{w} r) = P(l \wedge w \wedge r) = \frac{|E(l \wedge w \wedge r)|}{|E|} \quad (1)$$

$|E(l \wedge w \wedge r)|$ denotes the number of edges satisfying $l \wedge w \wedge r$. *Support* of $l \wedge w$ is defined as

$$supp(l \wedge w) = P(l \wedge w) = \frac{|E(l \wedge w)|}{|E|} \quad (2)$$

$|E(l \wedge w)|$ is the number of edges satisfying $l \wedge w$. □

With $|E|$ being a constant for a given network, we can use **absolute support** by dropping the denominator $|E|$. While support measures the *generality* of a GR, confidence measures the *strength* of a GR.

Definition 3: [Confidence] *Confidence* of $l \xrightarrow{w} r$ is defined as

$$conf(l \xrightarrow{w} r) = P(r \mid l \wedge w) = \frac{supp(l \xrightarrow{w} r)}{supp(l \wedge w)} \quad \square \quad (3)$$

B. Non-homophily Preference

The above support/confidence metric has been used in the literature to mine interesting association rules, by specifying a minimum threshold on support and a minimum threshold on confidence. However, many GRs that have a high support and a high confidence, like GR3, often are known and well expected because of the homophily effect, and those that do not follow from homophily but are still interesting, like GR4, are missed due to a low confidence. To unearth interesting GRs like GR4, the support/confidence metric approach has to set the thresholds for support and confidence at a very low level, leading to a much larger search space. In this paper we assume that the homophily principle is known and we are interested in GRs that are not expected from the homophily principle. Therefore, the confidence metric is not suitable for our purpose and we need a new metric to identify interesting GRs. First, let us clarify the notion of homophily.

Homophily attributes. Intuitively, a GR is considered to follow from the homophily principle if the LHS and RHS of this GR share a (set of) value(s). However, the homophily is attribute sensitive so that sharing values on certain attributes are not considered as trivial, such as the attribute SEX in a dating site. We differentiate an attribute as either a homophily attribute or a non-homophily attribute. An homophily attribute refers to an attribute on which the individuals sharing the same value are more likely to connect to each other. For a given social network, we assume that the setting of homophily attributes is specified. Some existing works, like [27], studied the methods to identify homophily attributes. In many cases, homophily attributes are known from a common sense. For example, EDU is likely a homophily attribute for dating relationships whereas SEX is a non-homophily attribute since dating could be between two people of same or opposite sex.

Trivial GRs. We say that a GR $l \xrightarrow{w} r$ is *trivial* if all of the values in r are from homophily attributes and $r \subseteq l$. A trivial GR is expected from the homophily principle, so we are only interested in non-trivial GRs.

To capture and rank the GRs not expected from homophily, we propose to exclude the homophily effect from confidence. Consider GR4, (SEX:*F*, EDU:*Grad*) $\xrightarrow{dates}$ (SEX:*M*, EDU:*College*), in Example 2. Assume that EDU is a homophily attribute. The confidence of GR4 is given by $supp(GR4)/supp(l \wedge dates) = 2/6$. 4 of the support $supp(l \wedge dates) = 6$ is contributed by the homophily effect represented by GR3 (SEX : *F*, EDU : *Grad*) $\xrightarrow{dates}$ (EDU : *Grad*), $supp(GR3) = 4$. Excluding this effect from $supp(l \wedge dates) = 6$, the new metric of GR4 is $2/(6-4) = 100\%$, read as: for women with *Grad* education, when not dating men having *Grad* education, they were dating men having *College* education with 100% probability.

In general, for a GR $l \xrightarrow{w} r$, let β denote the homophily attributes in $\mathcal{R}$ that occur in $\mathcal{L}$ but have *different* values in the two sides, i.e.,

$$\beta = \{A^r \in \mathcal{R} \mid A^l \in \mathcal{L}, r[A^r] \neq l[A^l]\} \quad (4)$$

Let $l[\beta]$ denote the condition for the RHS containing the values in l restricted to β. We define *homophily effect* as

$$l \xrightarrow{w} l[\beta] \quad (5)$$

In example GR4, EDU is the only homophily attribute and the values for EDU on both sides are different, thereby, β = {EDU} , and the homophily effect is (SEX:*F*, EDU:*Grad*) $\xrightarrow{dates}$ (EDU:*Grad*). Recall *conf* $(l \xrightarrow{w} r) = \frac{supp(l \xrightarrow{w} r)}{supp(l \wedge w)}$. We can exclude the homophily effect by subtracting $supp(l \xrightarrow{w} l[\beta])$ from the denominator $supp(l \wedge w)$ in the confidence. This gives rise to the following new metric.

Definition 4: [Non-homophily Preference] The definition of *non-homophily preference* of a GR $l \xrightarrow{w} r$ is given by

$$nhp(l \xrightarrow{w} r) = P(r \mid l \wedge w \wedge \neg l[\beta]) = \frac{supp(l \xrightarrow{w} r)}{supp(l \wedge w) - supp(l \xrightarrow{w} l[\beta])} \quad \square \quad (6)$$

Intuitively, $nhp(l \xrightarrow{w} r)$ is the conditional probability of links going to a node described by r, given that they satisfy $l \wedge w$ and do not go to a node described by $l[\beta]$.

Remark 1: In the case of $\beta = \emptyset$, the edges due to the homophily effect do not exist, we define $supp(l \xrightarrow{w} l[\beta]) = 0$; consequently, nhp degenerates to *conf*. Hence, *conf* is a special case of nhp where there is no homophily attribute. In the case of $\beta \neq \emptyset$, $nhp \geq$ *conf*, so excluding the homophily effect boosts the rank of a GR not expected from homophily. This is exactly what we want to achieve.

Theorem 1: Assume $supp(l \xrightarrow{w} r) \neq 0$. (i) The denominator in Eqn. (6) is not zero. (ii) $nhp \in [0, 1]$.

Proof: (i) If $\beta = \emptyset$, the denominator in Eqn. (6) is equal to $supp(l \wedge w)$, which is not equal to 0. Assume $\beta \neq \emptyset$. Suppose $supp(l \wedge w) - supp(l \xrightarrow{w} l[\beta]) = 0$, i.e., $supp(l \wedge w) =$

$supp(l \xrightarrow{w} l[\beta])$, this implies that all edges satisfying $l \wedge w$ go to the nodes covered by $l[\beta]$ and no edge goes to the nodes covered by r, i.e., $supp(l \xrightarrow{w} r) = 0$. But this contradicts the assumption.

(ii) If $\beta = \emptyset$, the denominator in Eqn. (6) is equal to $supp(l \wedge w)$, so nhp has a value in the range $[0, 1]$. If $\beta \neq \emptyset$, it suffices to note that the links accounted for by $supp(l \xrightarrow{w} r)$ and the links accounted for by $supp(l \xrightarrow{w} l[\beta])$ are disjoint (because r and l disagree on β), and both are subsets of those accounted for by $supp(l \wedge w)$. ■

C. Top-k GRs

Some GRs are interesting to users while some are not, we can use a threshold of support and non-homophily preference to select the interesting ones. Furthermore, for two GRs g_1: $l_1 \xrightarrow{w_1} r_1$ and g_2: $l_2 \xrightarrow{w_2} r_2$, if $l_1 \subseteq l_2$, $w_1 \subseteq w_2$, and $r_1 = r_2$, we say that g_1 is *more general* than g_2, and g_2 is *more special* than g_1. Intuitively, if g_1 is more general than g_2, g_1 is a similar tendency to g_2 but covers more nodes on LHS. In this case, if both g_1 and g_2 satisfy certain support and non-homophily preference thresholds, g_1 would make g_2 redundant.

On account of the above discussion, finding the k most interesting GRs offers a brief and valuable overview of the entire social network. Hence, this problem is formulated as follows.

Definition 5: [Top-k GRs] Given the homophily settings for attributes, a support threshold $minSupp$, a non-homophily preference threshold $minNhp$, and an integer k, a non-trivial $l \xrightarrow{w} r$ is a *top-k GR* if the three conditions hold:

- (1) $supp(l \xrightarrow{w} r) \geq minSupp$ and $nhp(l \xrightarrow{w} r) \geq minNhp$;
- (2) no non-trivial GR is more general than $l \xrightarrow{w} r$ while satisfying (1);
- (3) no more than $k-1$ non-trivial GRs have a higher rank while satisfying (1) and (2), where the rank is measured by non-homophily preference, followed by support, followed by the alphabetical order of GRs.

The objective is to mine the top-k GRs. □

IV. Mining Top-k GRs

One baseline algorithm for finding top-k $l \xrightarrow{w} r$ is to apply regular Apriori-like algorithms such as [5] to find frequent sets $l \wedge w$ and $l \wedge w \wedge r$ above the $minSupp$ threshold and then construct GRs in a post-processing step using the $minNhp$ threshold. This algorithm does not work well for GRs with a small support because there are too many frequent sets when $minNhp$ is small. In fact, strong social ties typically exist among small groups. Another issue is that frequent set mining usually requires collecting all information in one table. For graph data, this means replicating the node information for every edge adjacent to the node, and the size of this table is $|E| \times (2 \times \#Attr_V + \#Attr_E)$, where $\#Attr_V$ is the number of attributes in V and $\#Attr_E$ is the number of attributes in E. The term $|E| \times 2 \times \#Attr_V$ usually causes storage explosion and imposes a bottleneck for most graph algorithms, especially for high dimensional nodes with large $\#Attr_V$ and densely connected graphs with large $|E|$.

Another straightforward approach is to use a threshold for standard confidence (as defined in Definition 3), *minConf*, and $minSupp$ to mine all the GRs that satisfy these two criteria, then remove the trivial (homophilic) GRs in a post-processing phase to get the final results. This approach has the following drawbacks. First, as discussed in Section III-B, the confidence metric favors GRs that follow from the homophily principle so that the majority of the high-confidence GRs in the top-k results are trivial (we will show this in Table II). That is to say, many non-trivial and interesting GRs are not returned because either their *conf* are less than *minConf* or they are not ranked as the top k GRs. As a result, this algorithm has to set a very small *minConf* and very large k to first let the non-trivial GRs to be returned before doing the post-processing. By doing this, the efficiency of this approach becomes terrible owing to the computation of the huge number of trivial GRs. Second, the post-processing for the great number of trivial GRs is another cost, which makes this approach rather worse.

An ideal algorithm is that it examine only the necessary GRs and return the top-k results in one phase. To achieve this and address the issues mentioned above, the key is to push the $minNhp$ threshold, in addition to the $minSupp$ threshold, as early as possible. Besides, storing edge and node attributes information separately without duplication helps a lot. In this section, we first introduce the data model of representing the social networks that contain edge and node attributes information, then we mainly focus on a new search strategy for pushing the $minNhp$ threshold. The full implementation of our algorithm will be presented in Section IV.

A. Data Model

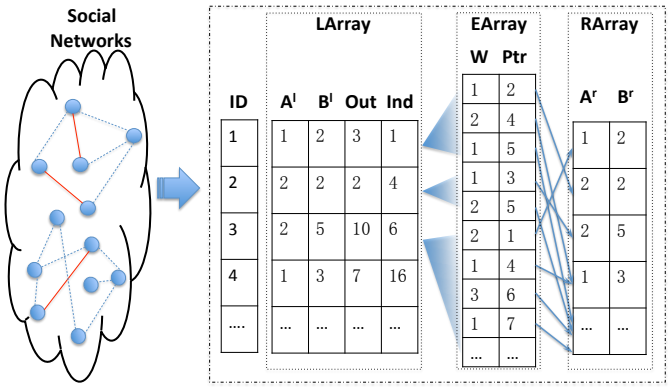


Fig. 2: Data model: LArray, EArray and RArray

For the sake of illustration, let's consider two node attributes A, B and one edge attribute W. For each node attribute A, we use the symbol A^l for the occurrence in LHS of a GR and use the symbol A^r for the occurrence in RHS. Then, we shall store the node and edge information of social networks separately as shown in Fig. 2.

LArray contains the records for individuals that could occur in the LHS of GRs and RArray contains the records for

individuals that could occur in the RHS of GRs. Out is the out-degree of a record and Ind is the starting position of the outgoing edges in EArray. EArray contains one record for each edge and Ptr is the pointer to the record for the destination node in RArray. We assume that this structure is held in memory and use it to partition the data for counting the support for GRs. For example, the first row in LArray represents the record 1 for LHS, which connects to the destination records 2, 4 and 5 for RHS, found by the pointers Ptr kept in the entries $[Ind, Ind+Out-1]$ of EArray. Note that RArray and LArray are for destinations and sources of edges (thus, not a subset of each other) and will be sorted by the different attributes for RHS and LHS for counting support. For this reason, RArray and LArray must be separately stored.

This compact data structure has the size $|V| \times (\#Attr_V + 2) + |E| \times (\#Attr_E + 1) + |V| \times \#Attr_V$, which eliminates the bottleneck term $|E| \times 2 \times \#Attr_V$ of the single table representation as mentioned above. The difference is usually large because $\#Attr_V$ is typically much larger than $\#Attr_E$ and a node typically connects to, or is connected from, multiple nodes. Even for a sparse network, the space requirement of the compact data model is also smaller since the nodes with zero out-degree of in-degree will not appear in LArray or RArray.

B. Pruning Strategies

To prune GRs using a minimum threshold on nhp (Definition 4), the challenge is that, shown in Theorem 2(2,3), nhp has anti-monotonicity only for "certain cases"; for the remaining cases, adding a value for a homophily attribute to RHS would increase or decrease nhp, so the traditional tree-based pattern enumeration cannot prune GRs using a threshold of nhp. See more discussion in Remark 2. In the rest of this section, we devise a new enumeration to manifest the anti-monotonicity of nhp in all cases (Theorem 3). This strategy allows us to prune GRs based on the threshold of nhp. First, the next theorem states pruning properties of GRs.

Theorem 2: (1) $supp(l \xrightarrow{w} r)$ is not increased by adding an attribute value to l or r or w. (2) If $\beta \neq \emptyset$, $nhp(l \xrightarrow{w} r)$ is not increased by adding a value to r. (3) If $\beta = \emptyset$, $nhp(l \xrightarrow{w} r)$ is not increased by adding a value to r for a non-homophily attribute or for a homophily attribute not occurring in l.

Proof: (1) follows from the anti-monotonicity of support. nhp is equal to $\frac{supp(l \xrightarrow{w} r)}{supp(l \wedge w) - supp(l \xrightarrow{w} l[\beta])}$ (Definition 4). Adding a value to r does not affect $supp(l \wedge w)$, and if $\beta \neq \emptyset$, never increases $supp(l \xrightarrow{w} l[\beta])$ and $supp(l \xrightarrow{w} r)$. This shows (2). If $\beta = \emptyset$, adding a value to r for a non-homophily attribute, or a homophily attribute not occurring in l, preserves $\beta = \emptyset$, thus, $supp(l \xrightarrow{w} l[\beta]) = 0$. Then (3) holds similarly as in (2). ■

Remark 2: Theorem 2(1) enables $supp$ based pruning of GRs, and Theorem 2(2,3) enables nhp based pruning when expanding the RHS r of a GR under certain cases. The remaining case is expanding a value to r for a homophily attribute that occurs in l when $\beta = \emptyset$. In this case, nhp does not have the anti-monotonicity. To see this, suppose that we add a value b^r for a homophily attribute B to r, where some value b^l of B^l, $b^l \neq b^r$, has already occurred in l. Before the addition, $\beta = \emptyset$, thus, $supp(l \xrightarrow{w} l[\beta]) = 0$, but after the addition, $\beta \neq \emptyset$ (see Eqn. (4)), so $supp(l \xrightarrow{w} l[\beta]) \neq 0$. This change may increase or decrease $nhp(l \xrightarrow{w} r)$. In this case, nhp does not have anti-monotonicity.

In the remainder of this section, we propose a careful order of enumerating GRs $l \xrightarrow{w} r$ so that GRs can be pruned based on nhp in all the cases of adding a value to r.

C. Subset-First Depth-First (SFDF) Enumeration

We use a tree structure to represent all GRs where each tree node represents a subset $\mathcal{LWR}$ (see Table I for the definition of $\mathcal{L}$, $\mathcal{W}$ and $\mathcal{R}$) and all the corresponding GRs $l \xrightarrow{w} r$. This tree structure is only a conceptual representation and is not stored in entirety. The nodes of this tree are enumerated to ensure two properties:

- *Property 1*: Enumerate a subset $\mathcal{LWR}$ by adding attributes in the order of those in $\mathcal{L}$, $\mathcal{W}$, and $\mathcal{R}$. This order enables the pruning in Theorem 2(1,2,3) where the values for r are added after those for l and w.
- *Property 2*: Enumerate a subset $\mathcal{L}_1\mathcal{W}_1\mathcal{R}_1$ before any subset $\mathcal{L}_2\mathcal{W}_2\mathcal{R}_2$ where $\mathcal{L}_1 \subseteq \mathcal{L}_2$, $\mathcal{W}_1 \subseteq \mathcal{W}_2$, and $\mathcal{R}_1 \subseteq \mathcal{R}_2$. This order ensures that the node for $l \xrightarrow{w} l[\beta]$ is enumerated before the node for $l \xrightarrow{w} r$ (because β is a subset of $\mathcal{R}$), hence, $supp(l \xrightarrow{w} l[\beta])$ was computed before computing $nhp(l \xrightarrow{w} r)$. This is necessary because the latter depends on the former.

The regular depth-first enumeration does not provide Property 2, and the regular breadth-first enumeration (level order) meets these requirements but has to keep all nodes and their GRs at the same level, which imposes a bottleneck on memory. We propose a novel strategy, called *subset-first depth-first* (SFDF), that will enumerate a subset before a superset like the breadth-first enumeration but is depth-first (to avoid the memory bottleneck).

To ensure that each subset $\mathcal{LWR}$ is enumerated at most once, we impose the following order on all attributes:

$$\tau : NH^r, H^r, W, NH^l, H^l \tag{7}$$

where NH^l denotes non-homophily attributes for LHS, and NH^r denotes non-homophily attributes for RHS. Similarly, H^l and H^r denote homophily attributes for LHS and RHS, respectively. W denotes edge attributes.

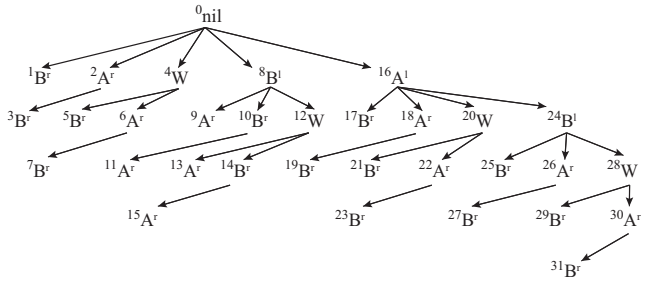


Fig. 3: Subset-First Depth-First enumeration

At any tree node t, $label(t)$ denotes the labeling attribute for t, $path(t)$ denotes the attribute set $\mathcal{LWR}$ constructed by

all the labels for the nodes on the path from the root to t, and $tail(t)$ denotes the prefix of the list τ to the left of the attribute $label(t)$. $tail(t)$ is the set of unused attributes that can be to expand $path(t)$ in the subtree below t. Initially, $label(root)$ is nil, $path(root)$ is empty, and $tail(root) = \tau$. If $tail(t) \neq \emptyset$, for each attribute in $tail(t)$ in order, t has one child t' labeled by the attribute. Note that this order will expand the subset $path(t)$ by adding the attributes in the order H^l, NH^l, W, H^r, NH^r, i.e., those for LHS, followed by those for edges, followed by those for RHS. This gives Property 1.

For the sake of illustration, we assume that both A and B are homophily attributes (the enumeration of non-homophily attributes are straightforward as discussed below). $NH^r = NH^l = \emptyset$, $H^r = \{B^r, A^r\}$, $H^l = \{B^l, A^l\}$, and $\tau = (B^r, A^r, W, B^l, A^l)$. Fig. 3 shows the SFDF enumeration of all subsets $\mathcal{LWR}$ with the order indicated by the sequence numbers aside the nodes. Let t_i denote the tree node numbered i. At the root t_0, $label(t_0) = nil$, $path(t_0) = \emptyset$, and $tail(t_0) = \tau$. The root has five child nodes, $t_1, t_2, t_4, t_8, t_{16}$, labeled B^r, A^r, W, B^l, A^l in that order. Next, the SFDF order enumerates t_1. $tail(t_1) = \emptyset$ and t_1 has no child. The next node enumerated is t_2 labeled A^r, $tail(t_2) = (B^r)$ and t_2 has one child t_3 labeled B^r. $path(t_3) = \{A^r, B^r\}$, which represents all GRs $l \xrightarrow{w} r$ with $\mathcal{L} = \emptyset$, $\mathcal{W} = \emptyset$, and $\mathcal{R} = \{A^r, B^r\}$. Similarly, t_4, t_5, t_6, t_7 are enumerated following this order.

At node t_8, $path(t_8) = \{B^l\}$ and $tail(t_8) = (B^r, A^r, W)$. For the first time, some homophily attribute, B, occurs in the LHS. This node represents the enumerated subset $\mathcal{LWR}$ where $\mathcal{L} = \{B^l\}$ and $\mathcal{W} = \mathcal{R} = \emptyset$. Note $\beta = \emptyset$. t_8 has three child nodes labeled B^r, A^r, W. Following the above order, the subset B^lB^r will be enumerated before the subset B^lA^r, then the subset $B^lA^rB^r$ will be enumerated as a child node of B^lA^r (by adding B^r). This is exactly the case discussed in Remark 2 where a homophily attribute B^l has a value in l and adding a new value for B^r to r changes $\beta = \emptyset$ to $\beta \neq \emptyset$, causing the lack of anti-monotonicity of nhp.

To avoid this problem, at node t_8, it helps to add A^r (because A^l does not occur in the LHS) *before* adding B^r (because B^l occurs in the LHS). In Fig. 3, this order ensures that the subset B^lA^r (at t_9) is enumerated before the subset B^lB^r (at t_{10}), therefore, the subset $B^lB^rA^r$ (at t_{11}) is enumerated as a child node of B^lB^r instead of a child node of B^lA^r. This is defined by the dynamic order of $tail(t)$ below.

Dynamic ordering of $tail(t)$. At any node t with $path(t) = \mathcal{LWR}$, let NH^l, NH^r, W, H^l, H^r be the same as in Eqn. (7). Let H_1^r and H_2^r be the partitioning of H^r, where H_1^r contains those A^r with the corresponding A^l not enumerated in $path(t)$ and H_2^r contains those A^r with A^l enumerated in $path(t)$. We *dynamically* order the attributes in $tail(t)$ at a node t as follows:

$$NH^r, H_1^r, H_2^r, W, NH^l, H^l \tag{8}$$

In other words, the homophily attributes in H^r are dynamically ordered on the basis of whether their corresponding attributes were enumerated in the LHS at t. Consequently, the attributes in H_2^r are added to $path(t)$ before the attributes in H_1^r.

Consider the node t_8 again. Recall that $path(t_8) = \{B^l\}$ and $tail(t_8) = (B^r, A^r, W)$. $H_1^r = \{A^r\}$ and $H_2^r = \{B^r\}$ because A^l was not enumerated in $path(t_8)$ and B^l was enumerated in $path(t_8)$. Then, $tail(t_8)$ is dynamically ordered as (A^r, B^r, W), instead of the static order (B^r, A^r, W). This order ensures that B^r is added to $path(t_8)$ before A^r if both B^r and A^r appear in the path, as shown by the path t_8, t_{10}, t_{11}. On the path t_4, t_6, t_7, B^r is added to $path(t_4)$ after A^r. This does not contradict our order because no homophily attribute was enumerated in $path(t_4)$, i.e., $H_1^r = \{B^r, A^r\}$ and $H_2^r = \emptyset$. The next theorem shows that this dynamical order restores the anti-monotonicity of nhp.

Theorem 3: Assume that $tail(t)$ is dynamically ordered at a node t described above, and g' and g are non-trivial GRs where g' is obtained from g by adding one or more values to the RHS of g. Then $nhp(g') \leq nhp(g)$.

Proof: If $\beta \neq \emptyset$ for g, Theorem 2(2) implies $nhp(g') \leq nhp(g)$. We assume $\beta = \emptyset$ for g. Let g' be the result of adding a value b to the RHS of g. If b is a value for an attribute in H_1^r or NH^r, Theorem 2(3) implies $nhp(g') \leq nhp(g)$. So we assume that b is a value for a homophily attribute in H_2^r. In this case, according to the dynamic ordering of $tail(t)$, the RHS of g contains only values for attributes in H_2^r since the values for attributes in H_1^r and NH^r are added after those for attributes in H_2^r. Thereby, all attributes in the RHS of g are homophily attributes and occur in the LHS. Then the assumption $\beta = \emptyset$ implies that the values of these attributes are contained in the LHS of g, hence, g is a trivial GR, contradicting our assumption. This shows that b cannot be a value for a homophily attribute in H_2^r if $\beta = \emptyset$. The case of adding more values to the RHS of g follows by repeating the above argument on g'. ■

The above enumeration order ensures that our depth-first traversal enumerates smaller subsets $\mathcal{LWR}$ before enumerating larger ones, i.e., Property 2, adds the attributes for LHS before adding the attributes for RHS, i.e., Property 1, and restores the anti-monotonicity of nhp, i.e., Theorem 3. All these properties are essential for pruning GRs based on the threshold of nhp.

D. Computing Non-homophily Preference

A remaining issue is how to compute nhp at a node. Suppose that we are enumerating the current node t for GRs $l \xrightarrow{w} r$. In $nhp(l \xrightarrow{w} r) = \frac{supp(l \xrightarrow{w} r)}{supp(l \wedge w) - supp(l \xrightarrow{w} l[\beta])}$ (Definition 4), $supp(l \xrightarrow{w} r)$ is computed at t and $supp(l \wedge w)$ was computed at an *earlier* node because the attribute set for $l \wedge w$ is a subset of the attribute set for $l \xrightarrow{w} r$ (i.e., Property 2). In the following discussion, we consider $supp(l \xrightarrow{w} l[\beta])$ and assume $\beta \neq \emptyset$, thus, $supp(l \xrightarrow{w} l[\beta]) \neq 0$. Note $\beta \subseteq \mathcal{R}$. There are two cases:

Case 1: If $\beta \subset \mathcal{R}$, the node for $l \xrightarrow{w} l[\beta]$ was enumerated at an *earlier* node because the attribute set for $l \xrightarrow{w} l[\beta]$ is a subset of the attribute set for $l \xrightarrow{w} r$. Note that $l \xrightarrow{w} l[\beta]$ is a trivial GR, its support can be easily computed.

Case 2: $\beta = \mathcal{R}$. In this case, $supp(l \xrightarrow{w} l[\beta])$ is computed at the *current* node t for $l \xrightarrow{w} r$. An example is a GR g at t_{27}: $(a_2, b_2) \rightarrow (a_1, b_1)$, where a_2, a_1 are different values for attribute A, and b_2, b_1 are different values for B. So $\beta = \{A^r, B^r\}$ and

$$nhp(g) = \frac{supp((a_2, b_2) \rightarrow (a_1, b_1))}{supp((a_2, b_2)) - supp((a_2, b_2) \rightarrow (a_2, b_2))} \tag{9}$$

Algorithm 1: GRMiner

```
1  Procedure Main()
2  initiate LArray, EArray and RArray;
3  RIGHT(RArray, tail(nil));
4  EDGE(EArray, tail(nil));
5  LEFT(LArray, tail(nil)));
6  Output(top[k]);
7  Procedure LEFT(data, Tail)
8  forall the dimension d both in Tail and in LArray do
9      forall the partition p of data on dimension d do
10         if supp(p) < minSupp then
11             return;
12         RIGHT(getRight(p), tail(p.Att));
13         EDGE(getEdge(p), tail(p.Att));
14         LEFT(p, tail(p.Att));
15 Procedure EDGE(data, Tail)
16 forall the dimension d both in Tail and in EArray do
17     forall the partition p of data on dimension d do
18         if supp(p) < minSupp then
19             return;
20         RIGHT(getRight(p), tail(p.Att));
21         EDGE(p, tail(p.Att));
22 Procedure RIGHT(data, Tail)
23 forall the dimension d both in Tail and in RArray do
24     forall the partition p of data on dimension d do
25         if supp(p) < minSupp OR nhp(p) < minNhp
           then
26             return;
27         if p is a non-trivial GR and no more general
           GR than p found then
28             update top[k] and minNhp if necessary;
29         RIGHT(p, tail(p.Att));
```

If we generate $(a_2, b_2) \rightarrow (a_2, b_2)$ before generating any other GRs with (a_2, b_2) on the LHS, $supp((a_2, b_2) \rightarrow (a_2, b_2))$ will be available when generating g. Enforcing this order only requires knowing the LHS of the current GR g, i.e., (a_2, b_2) in this example, therefore, can be easily implemented.

In both cases, $supp(l \xrightarrow{w} l[\beta])$ is either already computed or can be computed at the same node as for $l \xrightarrow{w} r$. Therefore, $nhp(l \xrightarrow{w} r)$ can be computed at the node for $l \xrightarrow{w} r$.

V. Algorithm Framework

We now present the algorithm framework, which partitions the data stored in the format as in Fig.2 and leverages the enumeration and pruning strategies presented in Section IV.

Our algorithm enumerates each attribute subset ($\mathcal{LWR}$) following the SFDF order as described in the last section. To compute $supp$ and nhp of the GRs at the node for $\mathcal{LWR}$, it partitions the data using the attribute set and then considers each partition recursively. A linear sorting method, Counting Sort [28], is adopted to sort and get the aggregate of each partition. It sorts in $O(N)$ time without any key comparisons. Our algorithm prunes further partitioning using the thresholds on $supp$ and nhp as in Theorem 2(1) and Theorem 3. We discussed how to compute nhp for a given partition in Section IV-D. Below, we focus on how to partition the data using LArray, EArray, and RArray as introduced in Section IV-A.

Algorithm 1, GRMiner, gives the pseudo-code of our algorithm. The main procedure starts with loading LArray, EArray and RArray into memory at line 2. tail() returns the attributes (dimensions) that will be used to expand the attribute set $\mathcal{LWR}$, similar to $tail(t)$ in Section IV-C. Initially, tail(nil) returns all the attributes in the order in Eqn. (8). In our running example, $tail(nil) = \{B^r, A^r, W, B^l, A^l\}$, where $\{B^r, A^r\}$ is in RArray, $\{W\}$ is in EArray, and $\{B^l, A^l\}$ is in LArray.

At the current node t of the tree, $data$ denotes the data partition generated by $\mathcal{LWR}$ at t. Since the attributes in $tail(t)$ are contained in the tables LArray, EArray, and RArray, we use three recursive procedures **RIGHT**$(data, Tail)$, **EDGE**$(data, Tail)$ and **LEFT**$(data, Tail)$ to partition $data$, where $Tail$ is a variable for $tail(t)$. Initially, $data$ is the entire tables LArray, EArray, and RArray and $Tail = tail(nil)$, at lines 3 - 5. Partitioning $data$ by an attribute in $tail(t)$ generates the partitions for a child node created as described in Section IV-C. These calls then search recursively deeper into the enumeration tree, explained below. On return from all calls, $top[k]$ contains the top-k GRs.

$LEFT(data, Tail)$ partitions $data$ using each dimension occurring both in $Tail$ and in LArray (line 8) (i.e., the dimensions in $Tail$ contained in LArray). By abuse of notation, for each partition p, we also use p to denote the corresponding GR. $supp(p)$ returns the support of p and $p.Att$ returns the attributes on which p has been partitioned. $p.Att$ corresponds to $path(t)$ in Section IV-C. If $supp(p) < minSupp$, the procedure returns immediately (line 11), otherwise, p is recursively partitioned on the next three lines. The functions $getRight(p)$ and $getEdge(p)$ expand the partition p to the records in RArray and EArray, respectively.

$EDGE(data, Tail)$ is similar to $LEFT(data, Tail)$ except that it partitions $data$ by each dimension occurring both in $Tail$ and in EArray, and recursively processes each partition p by the calls $RIGHT()$ and $EDGE()$.

$RIGHT(data, Tail)$ partitions $data$ by each dimension occurring both in $Tail$ and in RArray, and recurs on each partition. Line 25 checks if p meets the thresholds for support and non-homophily preference, and Line 27 checks if p represents a non-trivial GR and if a more general GR than the GR for p was generated before. Since our enumeration examines smaller subsets of attributes before examining larger subsets, once a GR passes this checking, no later GR can be more general than it, so every GR in $top[k]$ is a most general GR. Line 28 updates the $top[k]$ list if the GR for p is among the top k GRs so far, and upgrades $minNhp$ by the non-homophily preference of the least ranked GR in $top[k]$.

Theorem 4: (1) $top[k]$ returned by GRMiner contains the top-k GRs. (2) A non-trivial GR is examined by Algorithm 1 only if it passes both $minSupp$ and $minNhp$. □

The work of Algorithm 1 is proportional to the number of GRs examined. (2) implies that no time is spent on examining any non-trivial GRs that do not meet the thresholds $minSupp$

and $minNhp$, thanks to the checking at lines 10, 18, 25, and Theorem 3. Typically, much fewer GRs are examined because $minNhp$ is dynamically updated to the smallest non-homophily preference of the current top-k GRs (line 28). We will examine this effect of $minNhp$ on real life data sets in Section VI.

VI. Experiments

We evaluated the GRMiner algorithm on real life data on CentOS 6.4 with Intel 8-core processors 2.53GHz and 12G of RAM. The programs were written in C++.

A. Data Sets

We used two public real-world data sets: Pokec Social Network data[4] and DBLP co-authorship data[5] because the domains of these data sets are easy to understand, which is essential for interestingness studies.

Pokec Social Network Data. Pokec is the most popular online social network service in Slovakia for discovering, chatting and dating with online friends. This data set contains anonymized users with profile data and directed friendships between users. We extracted 6 most important node attributes: `Gender (G,3)`, `Age (A,11)`, `Region (R,188)`, `Education (E,10)`, `What-Looking-For (L,11)`, and `Marital Status (S,7)`, where the letter and number in a bracket are the abbreviation and domain size of an attribute. We specify A, R, E, L as homophily attributes. While all attributes have drop lists for choosing their values, E, L, S are also fillable with any text. We used the values from the drop list whenever they were chosen, and otherwise, the user-filled text subject to the following preprocessing in order: (1) Remove all characters except letters and apply standard IR pre-processing to the filled text. (2) For the words that occur in more than 200 user profiles, replace them by their English synonym and mark the other words as "invalid". (3) Use the highest level for E (for example, keep "*Master*" if both "*Bachelor*" and "*Master*" are filled); and for L and S, use the word with highest frequency. (4) Keep only user profiles containing no "invalid" value. The final induced graph has 1,436,515 (87.98%) users and 21,078,140 (68.83%) directed edges. In addition, we discretized the domain of `Age` into "0-6", "7-13", "14-17", "18-24", "25-34", "35-44", "45-54", "55-64", "65-79", and "80 or older".

DBLP Data. This is the co-authorship DBLP data set used in [1], and it contains 28,702 authors and 66,832 directed co-author relationships (we replace each undirected edge with two directed edges in opposite directions). Each author has two node attributes, `Area (A)` and `Productivity(P)`, and `Area` has 4 values *DB*, *DM*, *AI* and *IR*, and `Productivity` has 4 values *Poor*, *Fair*, *Good* and *Excellent*. We use the exact same criteria as in [1] to discretize the values for the two attributes. Definitely, an author may belong to multiple areas, we select one only among the four in which she/he publishes most. We specify `Area` as a homophily attribute since authors in the same areas tend to collaborate; while we specify `Productivity` as a non-homophily one, since it is common that students and professors are co-authors but generally students have much fewer publications than professors. We construct one edge attribute `Collaboration Strength (S)` with three domain values: *occasional* ($f = 1$), *moderate* ($2 \leq f < 5$), *often* ($f \geq 5$), where f is the number of papers co-authored by the two authors at the ends of an edge.

We evaluated the interestingness of GRs in Sections VI-B and VI-C, and evaluated the efficiency of the GRMiner algorithm in Section VI-D.

B. Interestingness Study for Pokec Data

One of our claims is that the proposed non-homophily preference metric (i.e., nhp) helps to identify interesting social ties beyond the well-known homophily principle. We evaluate this claim by comparing the top-k GRs ranked by nhp with the top-k GRs ranked by the standard confidence, *conf*. Note that when applying *conf*, homophily effect is not excluded. We set $minSupp = 0.1\%$ (i.e., absolute $minSupp$ = 21078), $minNhp$ and *minConf* at 50%, and k = 300. Table IIa shows the top-5 GRs ranked by nhp (in boldface) and top-5 GRs ranked by *conf*, plus one less ranked GR by nhp (the last row). 4 of the top-5 GRs ranked by *conf* are trivially expected from the homophily principle as both LHS and RHs contain the same value; this trend continues further down the list (not shown here). This suggests that the *conf* metric fails to find interesting relationships beyond what is known from the homophily effect. In contrast, the GRs ranked by nhp, i.e., P1-P5 and P207, tend to provide more insights. The *conf* of these GRs are included for comparison. These GRs are found because their nhp is high, even though their *conf* is low. Note that the proportion of data covered by a GR is captured by $supp$. We pick P2, P5, and P207 to discuss in details, other GRs are interpreted in a similar way.

P2: (E:*Basic*)→ (E:*Secondary*). This GR indicates that for people with *Basic* education, when not partnering with people with the same education as their own, they preferred (in 68.7% cases) those with *Secondary* education. With *Training* being closer to *Basic*, this GR is less expected from homophily of Education because *Training* is expected to be more popular among people with *Basic* education. Further examination of data reveals that the proportion of *Secondary* is 19.54% and the proportion of *Training* is only 1.9%, which is probably the reason for the high nhp of this GR.

P5: (L:*Sexual Partner*) → (G:*Female*). For this GR, nhp degenerates into *conf* because $\beta = \emptyset$ (no homophily attribute occurs on both sides). This GR suggests that for people describing themselves as looking for sexual partners, 64.7% of their partners are female. Starting with this GR and wondering whether gender has any impact on this behavior, we formed the following two hypothesis by varying P5, and queried their nhp and $supp$ from the data:

$$(G : \textit{Male}, L : \textit{Sexual Partner}) \rightarrow (G : \textit{Female})$$
$$nhp = 68.1\%; supp = 392652$$
$$(G : \textit{Female}, L : \textit{Sexual Partner}) \rightarrow (G : \textit{Male})$$
$$nhp = 48.8\%; supp = 71699$$

This pair suggests a big difference in the preference of opposite sex partners by males and females when looking for sexual

[4]http://snap.stanford.edu/data/soc-pokec.html
[5]http://dblp.uni-trier.de/xml/

TABLE II: Comparison of top GRs ranked by nhp and *conf*

(a) Pokec data set

Ranked by nhp	Ranked by *conf*
(L:*Chat*)→(L:*Good Friend*) P1: nhp = 69.5%; $supp$ = 649723 (*conf* = 30.9%)	(R:*27*)→(R:*27*) *conf* = 72.2%; $supp$ = 250930
(E:*Basic*)→(E:*Secondary*) P2: nhp = 68.7%; $supp$ = 682715 (*conf* = 15.4%)	(R:*24*)→(R:*24*) *conf* = 66.1%; $supp$ = 197374
(E:*Preschool*)→(E:*Basic*) P3: nhp = 66.1%; $supp$ = 54765 (*conf* = 30.4%)	(R:*32*)→(R:*32*) *conf* = 65.1%; $supp$ = 143219
(E:*Hardly Any*)→(E:*Basic*) P4: nhp = 65%; $supp$ = 34099 (*conf* = 30.7%)	(R:*10*)→(R:*10*) *conf* = 65%; $supp$ = 279623
(L:*Sexual Partner*) → (G:*Female*) P5: nhp = 64.7%; $supp$ = 468012 (*conf* = 64.7%)	(L:*Sexual Partner*) → (G:*Female*) *conf* = 64.7%; $supp$ = 468012
(G:*Male*, A:*25-34*) → (A:*18-24*) P207: nhp = 50.8%; $supp$ = 593785 (*conf* = 33.9%)	

(b) DBLP data set

Ranked by nhp	Ranked by *conf*
(A:*AI*)→(P:*Poor*) D1: nhp = 74.3%; $supp$ = 31330 (*conf* = 74.3%)	(A:*AI*)→(A:*AI*) *conf* = 88.8%; $supp$ = 37458
(A:*DB*)$\xrightarrow{often}$(A:*DM*) D2: nhp = 71.5%; $supp$ = 98 (*conf* = 6.98%)	(A:*DB*)→(A:*DB*) *conf* = 88.7%; $supp$ = 44980
(P:*Poor*)→(P:*Poor*) D3: nhp = 70.6%; $supp$ = 63174 (*conf* = 70.6%)	(A:*IR*)→(A:*IR*) *conf* = 75.9%; $supp$ = 16020
(P:*Excellent*)→(A:*DB*) D4: nhp = 68.1%; $supp$ = 2744 (*conf* = 68.1%)	(A:*AI*)→(P:*Poor*) *conf* = 74.3%; $supp$ = 31330
(A:*IR*)→(P:*Poor*) D5: nhp = 68.1%; $supp$ = 14368 (*conf* = 68.1%)	(A:*DM*)→(A:*DM*) *conf* = 72.3%; $supp$ = 14232
(A:*AI*, P:*Good*)→(A:*DM*) D16: nhp = 55.2%; $supp$ = 272 (*conf* = 11.6%)	

partners. Without first finding P5, it is difficult to find this difference from the collection of GRs.

P207: (G:*Male*, A:*25-34*) → (A:*18-24*). Again, we form hypothesis from the seed P207. We replace *Male* with *Female* on the LHS and get nhp = 32.8% and $supp$ = 204780, which suggests that women much less preferred younger partners than men. The next two variations show that this difference is even bigger for partner with opposite sex:

$$(G : \textit{Male}, A : \textit{25-34}) \rightarrow (G : \textit{Female}, A : \textit{18-24}) \quad nhp = 39.1\%; supp = 456201$$

$$(G : \textit{Female}, A : \textit{25-34}) \rightarrow (G : \textit{Male}, A : \textit{18-24}) \quad nhp = 12.8\%; supp = 80070$$

C. Interestingness Study for DBLP Data

For DBLP data, we set $minSupp = 0.1\%$ (i.e., absolute $minSupp$ = 67), $minNhp$ and *minConf* at 50%, and k = 20. Table IIb shows the top GRs ranked by nhp (in boldface) and *conf*. Similar to the study on Pokec Data, the top GRs ranked by nhp are more interesting than those ranked by *conf*. Recall that `Area (A)` is a homophily attribute and `Productivity(P)` is not.

D1 & D3 & D5: On surface, D1 & D3 & D5 suggests the preference to authors with *Poor* productivity. This is interesting as it contradicts with the common sense. A quick check on the data (by examining the values distribution on the attribute) tells that 91.18% of the authors have the value *Poor* for `P` because many authors are students and most co-authorship is between supervisors and students.

D2: (A:*DB*)$\xrightarrow{often}$(A:*DM*) D2 suggests that authors in the *DB* area often collaborate with those in the *DM* area when collaborating with those not in their own area. D16 is a similar pattern for authors in *AI* area. In fact, *DM* has the least proportion among all areas. Therefore, these GRs represent a true preference to *DM*, not due to data skewness. A possible reason is that *DM* is an interdisciplinary field that intersects database and machine learning (a subarea of AI).

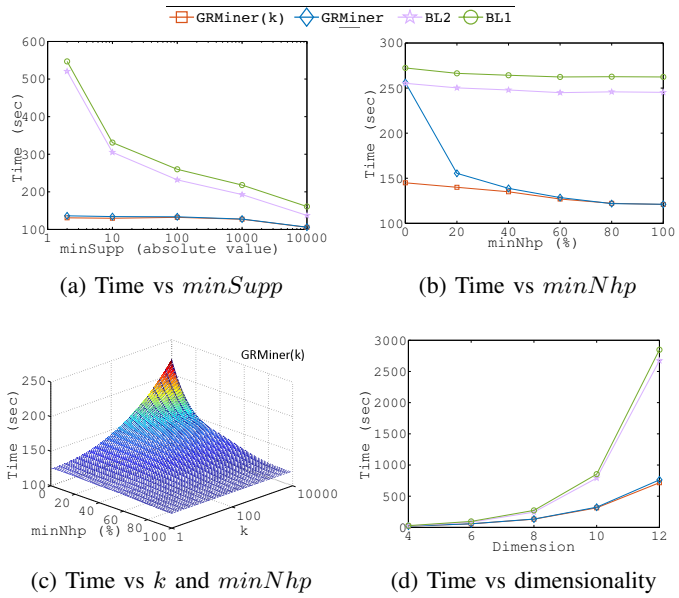
Remark 3: Finding top-k GRs typically serves the *entry point* in pattern mining. In the above case studies, the human analyst starts with top-k GRs found, forms new hypothesis through varying the GRs found, and compares such hypothesis as well as data distribution. This process can apply to the new hypothesis recursively. This cycle of hypothesis formulation and hypothesis comparison often leads to new insights into the behaviors of different groups of actors or an explanation of the presence of a GR. Unlike manual probing of a data set, top-k GRs provide an entry point to this cycle by filtering many uninteresting and trivial patterns.

D. Efficiency of Algorithms

Our algorithm finished running on the DBLP data set in no more than 0.483 seconds for all parameter settings. Therefore, our study below focuses on the Pokec data, which is much larger than the DBLP data. GRMiner(k) denotes the algorithm that pushes all the constraints of $minSupp$, $minNhp$, top-k, and generality of GRs to prune search space, as described in Section VI-D. GRMiner pushes all constraints except for the top-k constraint. The difference will tell the effectiveness of dynamically upgrading $minNhp$ to that of top-k GRs.

We consider two baseline solutions. One stores the node and edge attributes information in a single table, applies the BUC algorithm [23] to mine the combinations of attribute values above the threshold $minSupp$. We denote this baseline by BL1. The second baseline, BL2, is similar to BL1 but works with the node and edge attributes information separately stored in three tables. Both baselines prune the search space using the anti-monotonicity of support, but not $minNhp$, and find the top-k GRs in a post-processing step.

Unless otherwise stated, we consider the four node attributes with largest domain sizes, i.e., `Age`, `Region`,



(a) Time vs $minSupp$ (b) Time vs $minNhp$

(c) Time vs k and $minNhp$ (d) Time vs dimensionality

Fig. 4: Runtime for mining GRs for Pokec data

`Education` and `What-looking-for`, for examining various parameter settings. So the dimensionality of search space for GRs is 8. We set the ranges of (absolute) $minSupp$, $minNhp$, and k to $[2, 10000]$, [0%,100%], [1,10000], respectively, with the default settings 50, 50%, and 100. Fig. 4 summarizes the comparison on runtime of all algorithms.

minSupp. Fig. 4a presents runtime vs $minSupp$. For a small $minSupp$, the runtime of BL1 and BL2 increases quickly while the runtime of GRMiner(k) and GRMiner remains relatively stable, even when $minSupp$ reduces to 2. The efficiency of GRMiner(k) and GRMiner in the case of a small $minSupp$ comes from the fact that these algorithms prune the search space using $minNhp$. This is a huge advantage because a small $minSupp$ is often required for finding GRs with a high non-homophily preference that typically exist between small user groups.

minNhp and ***Top-k***. Fig. 4b studies the effect of $minNhp$. BL1 and BL2 do not benefit from a larger $minNhp$ since they employ only $minSupp$ for pruning. GRMiner(k) and GRMiner are significantly faster thanks to the $minNhp$ based pruning. For a small $minNhp$, GRMiner(k) is faster than GRMiner by dynamically upgrading $minNhp$ to the smallest nhp of the top-k GRs found. Fig. 4c examines the joint effect of k and $minNhp$ on GRMiner(k). As long as one of the two constraints is tight, i.e., a small k or a large $minNhp$, the pruning is effective. With a small k, the smallest nhp of top-k GRs is likely high, so the upgraded $minNhp$ has a similar effect to having a large user-specified $minNhp$.

Dimensionality. Fig. 4d shows the effect of the dimensionality $2l$, when the first l node attributes listed in Section VI-A are included and l varies from 2 to 6. All other parameters are set to their default settings. As the data has more node attributes, the runtime for GRMiner(k) and GRMiner increases much slower than the two baselines. This is because, as more attributes can occur on RHS, there is more room for $minNhp$ pruning in GRMiner(k) and GRMiner according to Theorem 3.

VII. Extensions

While non-homophily preference (nhp) is defined for the problem of mining GRs beyond homophily in this paper, the algorithm framework in Section V can be extended to different interestingness metrics to solve different tasks. The support-confidence metric has some drawbacks and several alternatives have been suggested to address these drawbacks in the literature. See [29], [30] for a discussion and motivation of such alternatives. The following are several examples of such alternative metrics after being adopted to a GR $l \xrightarrow{w} r$:

$$laplace(l \xrightarrow{w} r) = \frac{supp(l \xrightarrow{w} r) + 1}{supp(l \wedge w) + k} \tag{10}$$

where k is an integer greater than 1.

$$gain(l \xrightarrow{w} r) = supp(l \xrightarrow{w} r) - \theta \times supp(l \wedge w) \tag{11}$$

where θ is a a fractional constant between 0 and 1.

$$Piatetsky\_Shapiro(l \xrightarrow{w} r) = supp(l \xrightarrow{w} r) - \frac{supp(l \wedge w)supp(r)}{|E|} \tag{12}$$

$$conviction(l \xrightarrow{w} r) = \frac{|E| - supp(r)}{|E|(1 - conf(l \xrightarrow{w} r))} \tag{13}$$

$$lift(l \xrightarrow{w} r) = \frac{|E|conf(l \xrightarrow{w} r)}{supp(r)} \tag{14}$$

For example, a GR, $l \xrightarrow{w} r$, has a high confidence, but the true reason for this is that the relevant attribute value on RHS has a high population among all the edges, i.e., $supp(r)$ or $conf(\emptyset \xrightarrow{w} r) = \frac{supp(r)}{|E|}$ is high. One example is the GR D1 found in Section VI-C, which does not represent an interesting pattern. The *lift* metric, defined in Eqn. (14), can reduce the influence of this data skewness.

To adopt these alternative metrics for our algorithms for mining interesting GRs, a key observation is that all the above alternative metrics are defined using three supports, namely, $supp(l \xrightarrow{w} r)$, $supp(l \wedge w)$, and $supp(r)$, and these supports are easily computed. Therefore, in principle, the algorithm for mining top-k GRs presented in this paper can be applied as well if the nhp is replaced with one of the above alternative metrics. In addition, for $laplace$ or $gain$, the anti-monotonicity remains valid (proof omitted). This means that similar to the regular confidence based pruning, candidate GRs can be pruned based on a given threshold on $laplace$ or $gain$. For $Piatetsky\_Shapiro$, $conviction$, and $lift$, the corresponding pruning is not available because these metrics do not have the anti-monotonicity with respect to the RHS r, but the support based pruning is still applicable. For such metrics, the top-k GRs have to be found in a post-processing step after finding all the GRs satisfying the threshold on support.

VIII. Conclusion

Given the homophily observed on social interactions, we considered the problem of mining interesting interaction patterns that are *not* expected from homophily by excluding the impact of homophily from the interestingness metrics of social ties. We motivated and formulated this problem as mining top-k group relationships from a social network with respect to a specification of homophily attributes. We presented an efficient solution to this problem with a focus on pushing the new interestingness metric to prune the search space. We consider finding top-k group relationships as the start of analysis, not the end. However, this starting step is important as it provides the user with some sorts of leads to start with. Our empirical study on two real data sets demonstrated the potential of this approach in finding interesting social patterns.

Acknowledgement

The work of the second author is partially supported by a Discovery Grant from Natural Sciences and Engineering Research Council of Canada, partially done during a visit to Singapore Management University and a visit to SA Center for Big Data Research hosted in Renmin University of China. The SA Center is partially funded by a Chinese National 111 Project "Attracting International Talents in Data Engineering and Knowledge Engineering Research". The work of the third author is partially supported by the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office, Media Development Authority (MDA).

References

[1] P. Zhao, X. Li, D. Xin, and J. Han, "Graph cube: on warehousing and olap multidimensional networks," in *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*. ACM, 2011, pp. 853–864.

[2] M. McPherson, L. Smith-Lovin, and J. M. Cook, "Birds of a feather: Homophily in social networks," *Annual review of sociology*, pp. 415–444, 2001.

[3] T. Jambor and J. Wang, "Optimizing multiple objectives in collaborative filtering," in *Proceedings of the fourth ACM conference on Recommender systems*. ACM, 2010, pp. 55–62.

[4] J. McAuley, R. Pandey, and J. Leskovec, "Inferring networks of substitutable and complementary products," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2015, pp. 785–794.

[5] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," in *ACM SIGMOD Record*, vol. 22, no. 2. ACM, 1993, pp. 207–216.

[6] E. M. Rogers, *Diffusion of innovations*. Simon and Schuster, 2010.

[7] D. Chakrabarti and C. Faloutsos, "Graph mining: Laws, generators, and algorithms," *ACM Computing Surveys (CSUR)*, vol. 38, no. 1, p. 2, 2006.

[8] M. E. Newman, "The structure and function of complex networks," *SIAM review*, vol. 45, no. 2, pp. 167–256, 2003.

[9] A. Lancichinetti and S. Fortunato, "Community detection algorithms: a comparative analysis," *Physical review E*, vol. 80, no. 5, p. 056117, 2009.

[10] X. Xu, N. Yuruk, Z. Feng, and T. A. Schweiger, "Scan: a structural clustering algorithm for networks," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2007, pp. 824–833.

[11] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon, "Network motifs: simple building blocks of complex networks," *Science*, vol. 298, no. 5594, pp. 824–827, 2002.

[12] M. Kuramochi and G. Karypis, "Frequent subgraph discovery," in *Proceedings of the 2001 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2001, pp. 313–320.

[13] X. Yan and J. Han, "gspan: Graph-based substructure pattern mining," in *Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2002, pp. 721–724.

[14] L. Getoor and C. P. Diehl, "Link mining: a survey," *ACM SIGKDD Explorations Newsletter*, vol. 7, no. 2, pp. 3–12, 2005.

[15] J. Yang, J. McAuley, and J. Leskovec, "Community detection in networks with node attributes," in *Proceedings of the 13th IEEE international conference on Data Mining (ICDM),*. IEEE, 2013, pp. 1151–1156.

[16] J. J. Pfeiffer III, S. Moreno, T. La Fond, J. Neville, and B. Gallagher, "Attributed graph models: Modeling network structure with correlated attributes," in *Proceedings of the 23rd international conference on World wide web*. ACM, 2014, pp. 831–842.

[17] M. Kim and J. Leskovec, "Modeling social networks with node attributes using the multiplicative attribute graph model," *arXiv preprint arXiv:1106.5053*, 2011.

[18] W. Gatterbauer, S. Günnemann, D. Koutra, and C. Faloutsos, "Linearized and single-pass belief propagation," *Proceedings of the VLDB Endowment*, vol. 8, no. 5, pp. 581–592, 2015.

[19] Y. Tian, R. A. Hankins, and J. M. Patel, "Efficient aggregation for graph summarization," in *Proceedings of the 2008 ACM SIGMOD international conference on management of data*. ACM, 2008, pp. 567–580.

[20] C. Chen, X. Yan, F. Zhu, J. Han, and P. S. Yu, "Graph olap: Towards online analytical processing on graphs," in *Proceedings of the Eighth IEEE International Conference on Data Mining (ICDM'08)*. IEEE, 2008, pp. 103–112.

[21] Q. Qu, F. Zhu, X. Yan, J. Han, S. Y. Philip, and H. Li, "Efficient topological olap on information networks," in *Database Systems for Advanced Applications*. Springer, 2011, pp. 389–403.

[22] M. Berlingerio, F. Pinelli, and F. Calabrese, "Abacus: frequent pattern mining-based community discovery in multidimensional networks," *Data Mining and Knowledge Discovery*, vol. 27, no. 3, pp. 294–320, 2013.

[23] K. Beyer and R. Ramakrishnan, "Bottom-up computation of sparse and iceberg cube," in *ACM SIGMOD Record*, vol. 28, no. 2. ACM, 1999, pp. 359–370.

[24] G. I. Webb and J. Vreeken, "Efficient discovery of the most interesting associations," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 8, no. 3, p. 15, 2014.

[25] L. Dehaspe and H. Toivonen, "Discovery of frequent datalog patterns," *Data Mining and knowledge discovery*, vol. 3, no. 1, pp. 7–36, 1999.

[26] K. Wang, Y. Jiang, and L. V. Lakshmanan, "Mining unexpected rules by pushing user dynamics," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2003, pp. 246–255.

[27] A. L. Traud, P. J. Mucha, and M. A. Porter, "Social structure of facebook networks," *Physica A: Statistical Mechanics and its Applications*, vol. 391, no. 16, pp. 4165–4180, 2012.

[28] T. H. Cormen, *"8.2 Counting Sort", Introduction to algorithms (2nd ed.)*. MIT press, 2009.

[29] R. J. Bayardo Jr and R. Agrawal, "Mining the most interesting rules," in *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1999, pp. 145–154.

[30] J. Li, "On optimal rule discovery," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 4, pp. 460–471, 2006.