Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 35 (2014) 223 – 232

Procedia Computer Science

18th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems - KES 2014

A tree-based algorithm for mining diverse social entities

Peter Braun[a], Alfredo Cuzzocrea[b,*], Carson K. Leung[a,*], Richard Kyle MacKinnon[a], Syed K. Tanbeer[a]

[a]*Department of Computer Science, University of Manitoba, Winnipeg, MB, R3T 2N2, Canada*
[b]*ICAR-CNR and University of Calabria, Via P. Bucci, 41C, I-87036, Rende (CS), Italy*

Abstract

DiSE-growth, a tree-based (pattern-growth) algorithm for mining DIverse Social Entities, is proposed and experimentally assessed in this paper. The algorithm makes use of a specialized data structure, called *DiSE-tree*, for effectively and efficiently representing relevant information on diverse social entities while successfully supporting the mining phase. Diverse entities are popular in a wide spectrum of application scenarios, ranging from linked Web data to Semantic Web and social networks. In all these application scenarios, it has become important to analyze high volumes of valuable linked data and discover those diverse social entities. We complement our analytical contributions by means of an experimental evaluation that clearly shows the benefits of our tree-based diverse social entity mining algorithm.

Peer-review under responsibility of KES International.

Keywords: Data mining; diverse friends; friendship patterns; intelligent information & engineering systems; knowledge based and expert systems; social computing systems; social network analysis

1. Introduction

As technology advances, high volumes of valuable data (e.g., blogs, forums, wikis, and users' reviews) can be easily generated or collected from various data sources. These data are often related or linked, and thus form a web of linked data[4]. Over the few years, researchers have modelled, queried, and reasoned these linked web data.

In general, a social web is an instance of a web of linked data. Such a social web can be viewed as a collection of social relationships that link social entities (e.g., users). In recent years, researchers have exploited the social perspective or social phenomena in these webs of linked data[8,29]. Intuitively, social networks are made of social entities who are linked by some specific types of relationships (e.g., friendship, common interest, kinship). Facebook, Google+, LinkedIn, Twitter and Weibo[1,22,32] are some examples of social networks. Within these networks, a user f_i usually can create a personal profile, add other users as friends, endorse their skills/expertise, and exchange messages among friends. These social networks may consist of thousands or millions of users; each user f_i can have different number of friends. Among them, some are more important (or influential, prominent, and/or active) in a wide range

* Corresponding authors.
E-mail address: cuzzocrea@si.deis.unical.it (A. Cuzzocrea), kleung@cs.umanitoba.ca (C.K. Leung)

Peer-review under responsibility of KES International.
doi:10.1016/j.procs.2014.08.102

of domains than others[3,5,14,15]. Recognizing these diverse users can provide valuable information for various real-life applications when analyzing and mining high volumes of valuable social network data.

Some data mining techniques[13,31,35] have been developed over the past few years to help users discover implicit, previously unknown, and potentially useful information about the important friends or social entities. These techniques helps discover significant friends[25] or strong friends[26] based on the degree of one-to-one interactions (e.g., based on the number of postings to a friend's wall) in social networks.

In addition, it is also important to discover users who (i) are influential in the social networks, (ii) have high level of expertise in some domains, and/or (iii) have diverse interest in multiple domains. In other words, users may want to find important friends based on their influence, prominence, and/or diversity. For instance, some users may be narrowly interested in one specific domain (e.g., computers). Other users may be interested in a wide range of domains (e.g., arts, computers, sports), but their expertise level may vary from one domain to another (e.g., a user f_i may be a computer expert but only a beginner in sports). Hence, in this paper, we propose a tree-based mining algorithm to find from social networks those diverse users (i.e., diverse social entities) who are highly influential across multiple social network domains. To this end, one of our *key contributions* is an efficient tree-based (pattern-growth) algorithm called *DiSE-growth* for mining DIverse Social Entities from social networks. DiSE-growth takes into account multiple properties (e.g., influence, prominence, and/or diversity) of users in the networks. Another *key contribution* is a prefix-tree based structure called *DiSE-tree* for capturing social network data in a memory-efficient manner. Once the DiSE-tree is constructed, DiSE-growth computes the diversity of users based on both their influence and prominence to mine diverse groups of social entities.

The remainder of this paper is organized as follows. The next section presents related works. Section 3 introduces the notion of diverse social entities. Section 4 presents our DiSE-growth algorithm, which mines diverse social entities from our DiSE-tree. Experimental results are reported in Section 5. Conclusions and future work are given in Section 6.

2. Related works

The rapid growth and exponential use of social digital media over the past few years has led to an increase in popularity of social networks and the emergence of social computing[12,24,28]. Several data mining techniques[18,23,33] have been developed to help users extract implicit, previously unknown, and potentially useful information from linked web data and/or social network data such as blogs, forums, and wikis. For instance, researchers have modelled, queried, and reasoned about these linked web and/or social data. Most of these works focus on some specific data mining tasks. For example, Pernelle and Saïs[21] focused on classification rule learning for linked data. Ferrara et al.[10] proposed a feature-based approach to classify linked data. Besides the data mining task of classification, researchers have also examined relevant problems of detecting communities over social and information networks[6,7]. Furthermore, researchers have also examined other data mining tasks including clustering of social media data[27,30], mining and analysis of co-authorship networks[17,20], and visualization of social networks[9,16]. This paper, on the other hand, focuses on a different but also important aspect—namely, pattern mining on social networks.

Recent works on *pattern mining on social networks* also include the discovery of significant friends[25] and strong friends[26] based on the degree of one-to-one interactions (e.g., based on the number of postings to a friend's wall). However, there are situations in which one may want to find friends based on their relevant information (e.g., status of a friend in a social network) other than the number of messages or wall postings. For instance, a Facebook user may want to identify those prominent friends who have high impact (e.g., in terms of knowledge or expertise about a subject matter) in the social network. As another example, a LinkedIn user may want to get introduced to those second-degree connections who have rich experience in some profession. Similarly, a Twitter user may also be interested in following (and subscribing to a Twitter feed from) those who are highly diverse in multiple domains in the entire network. Hence, it is desirable to discover diverse social entities from the social network.

3. Fundamental concepts: diverse entities in social networks

To understand the concept of diverse entities in social networks, let us consider a social network on three different domains (domains D_1, D_2, D_3) and seven individuals—Aleksy, Bolek, Cyryl, Danuta, Edyta, Felicja, and Gustaw—

with prominence values in each domain, as shown in Table 1. Each *domain* represents a sub-category (e.g., arts, computer, sports) of interest. The *prominence value* of an individual reveals his level of expertise (e.g., importance, weight, value, reputation, belief, position, status, or significance) in a domain. In other words, the prominence value indicates how important, valued, significant, or well-positioned the individual is in each domain. The prominence value can be measured by using a common scale, which could be (i) specified by users or (ii) automatically calculated based on some user-centric parameters (e.g., connectivity, centrality, expertise in the domain, years of membership in the domain, degree of involvement in activities in the domain, numbers of involved activities in the domain). In this paper, the prominence value is normalized into the range (0, 1]. As the same individual may have different levels of expertise in different domains, his corresponding prominence value may vary from one domain to another. For instance, prominence value $Prom_{D_1}$(Aleksy) of Aleksy in domain D_1 is 0.45, which is different from $Prom_{D_2}$(Aleksy) = 0.60. Moreover, $Prom_{D_1}$(Aleksy) is higher than $Prom_{D_1}$(Cyryl) = 0.20, implying that Aleksy is more influential than Cyryl in domain D_1.

Like the existing settings of a social network[13,19], let $F = \{f_1, f_2, \ldots, f_m\}$ be a set of social entities/friends in a social network. An *interest-group list* $L \subseteq F$ is a list of individuals who are connected as friends due to some common interests (e.g., interested in ballet, ..., soccer). Let $G = \{f_1, f_2, \ldots, f_k\} \subseteq F$ be a *group of social entities* (i.e., friend group) with k friends. Then, $|G| = k$, which represents the number of individuals in G. A *friend network* $F_{SN} = \{L_1, L_2, \ldots, L_n\}$ is the set of all n interest-group lists in the entire social network. These lists belong to some domains (e.g., arts, computer, sports), and each domain contains at least one list. The set of lists in a particular domain D is called a *domain database* (denoted as F_D). Here, we assume that there exists an interest-group list in every domain. The *projected list* F_D^G of G in F_D is the collection of interest-group lists in F_D that contains social entity group G. The frequency $Freq_D(G)$ of G in F_D indicates the number of lists L_j's in F_D^G, and the frequencies of G in multiple domains are represented as $Freq_{D_{1,2,\ldots,d}}(G) = \langle Freq_{D_1}(G), Freq_{D_2}(G), \ldots, Freq_{D_d}(G)\rangle$.

For example, consider F_{SN} shown in Table 2, which consists of n=10 interest-group lists L_1, ..., L_{10} for m=7 social individuals/friends in Table 1. Each row in the table represents the list of an interest group. These 10 interest groups are distributed into d=3 domains D_1, D_2, and D_3. For instance, $F_{D_1} = \{L_1, L_2, L_3\}$. For group G = {Cyryl, Edyta}, its size $|G|$=2. As its projected lists on the 3 domains are (i) $F_{D_1}^G = \emptyset$, (ii) $F_{D_2}^G = \{L_5, L_7\}$ and (iii) $F_{D_3}^G = \{L_9\}$, its frequencies $Freq_{D_{1,2,3}}(G) = \langle 0, 2, 1\rangle$.

Table 1. Prominence of friends.

Friend (f_i)	Prominence $Prom(f_i)$ in arts (domain D_1)	$Prom(f_i)$ in computer (domain D_2)	$Prom(f_i)$ in sports (domain D_3)
Aleksy	0.45	0.60	0.50
Bolek	0.90	0.70	0.30
Cyryl	0.20	0.60	0.70
Danuta	0.30	0.50	0.40
Edyta	0.50	0.40	0.45
Felicja	0.42	0.24	0.70
Gustaw	0.57	0.10	0.20

Table 2. Lists of interest groups in F_{SN}.

Domain	Interest-group list L_j
D_1: arts	L_1 on ballet = {Aleksy, Bolek} L_2 on concerts = {Aleksy, Bolek, Danuta} L_3 on plays = {Cyryl, Danuta}
D_2: computer	L_4 on databases = {Bolek, Cyryl, Danuta} L_5 on data mining = {Bolek, Cyryl, Edyta} L_6 on knowledge-based systems = {Bolek, Gustaw} L_7 on social network analysis = {Cyryl, Edyta}
D_3: sports	L_8 on hockey = {Aleksy, Cyryl} L_9 on lacrosse = {Aleksy, Cyryl, Edyta} L_{10} on soccer = {Aleksy, Felicja}

Definition 1. The *prominence value* $Prom_D(G)$ of a friend group G in a single domain D is defined as the average of all prominence values for all the friends in G:

$$Prom_D(G) = \frac{\sum_{i=1}^{|G|} Prom_D(f_i)}{|G|}, \tag{1}$$

where $|G|$ is the size of G (i.e., the number of social individuals in G). Then, prominence values $Prom_{D_{1,2,\ldots,d}}(G)$ of a friend group G in multiple domains are represented as $Prom_{D_{1,2,\ldots,d}}(G) = \langle Prom_{D_1}(G), Prom_{D_2}(G), \ldots, Prom_{D_d}(G)\rangle$. □

Example 1. Consider F_{SN} shown in Table 2. The prominence value of friend group G = {Cyryl, Edyta} in D_1 = $\frac{Prom_{D_1}(\text{Cyryl})+Prom_{D_1}(\text{Edyta})}{|G|} = \frac{0.20+0.50}{2} = 0.35$. We apply similar computation on the other two domains D_2 and D_3 to get $Prom_{D_{1,2,3}}(G) = \langle 0.35, \frac{0.60+0.40}{2}, \frac{0.70+0.45}{2}\rangle = \langle 0.35, 0.5, 0.575\rangle$. □

Definition 2. The *influence* $Inf_D(G)$ of a group G of social entities/friends in a domain D in F_D is defined as the product of the prominence value of G in the domain D and its frequency in the domain database F_D, i.e.,

$$Inf_D(G) = Prom_D(G) \times Freq_D(G). \tag{2}$$

The influence $Inf_{D_{1,2,\ldots,d}}(G)$ of G in multiple domains is then represented as $Inf_{D_{1,2,\ldots,d}}(G) = \langle Inf_{D_1}(G), Inf_{D_2}(G), \ldots, Inf_{D_d}(G)\rangle$ because it is the "dot product" of (i) $Prom_{D_{1,2,\ldots,d}}(G) = \langle Prom_{D_1}(G), Prom_{D_2}(G), \ldots, Prom_{D_d}(G)\rangle$ and (ii) $Freq_{D_{1,2,\ldots,d}}(G) = \langle Freq_{D_1}(G), Freq_{D_2}(G), \ldots, Freq_{D_d}(G)\rangle$. □

Example 2. Recall from Example 1 that $Prom_{D_{1,2,3}}(G) = \langle 0.35, 0.5, 0.575\rangle$. As $Freq_{D_{1,2,3}}(G) = \langle 0, 2, 1\rangle$, the overall influence of G in all 3 domains can be calculated as $Inf_{D_{1,2,3}}(G) = \langle 0.35 \times 0, 0.5 \times 2, 0.575 \times 1\rangle = \langle 0, 1, 0.575\rangle$. □

Definition 3. The *diversity* $Div(G)$ of a group G of social entities/friends among all d domains in F_{SN} is defined as the average of all the influence values of G in all domains in the social network:

$$Div(G) = \frac{\sum_{j=1}^{d} Inf_{D_j}(G)}{d}, \tag{3}$$

where $Inf_{D_j}(G)$ is the influence of G in a domain D_j. □

Example 3. Continue with Example 2. Recall that $Inf_{D_{1,2,3}}(G) = \langle 0, 1, 0.575\rangle$. Then, the diversity of G in these d=3 domains in F_{SN} is $Div(G)$=$\frac{0+1+0.575}{3}$=0.525. □

Definition 4. A group G of social entities in a social network F_{SN} is considered *diverse* if its diversity value $Div(G) \geq$ user-specified minimum threshold *minDiv*, which can be expressed as an absolute (non-negative real) number or a relative percentage (with respect to the size of F_{SN}).

Given (i) F_{SN} and (ii) *minDiv*, the research problem of *mining diverse social entities from social networks* is to find every group G of friends having $Div(G) \geq minDiv$. □

Example 4. Let group G = {Cyryl, Edyta}. Recall from Example 3 that diversity $Div(G) = 0.525$. Given (i) F_{SN} in Table 2 and (ii) the user-specified *minDiv*=0.5, G is *diverse* because $Div(G)$=0.525 ≥ 0.5=*minDiv*.

However, group G' = {Edyta}, such that $G' \subseteq G$ is *not* diverse because $Div(G') = \frac{(0.5\times0)+(0.4\times2)+(0.45\times1)}{3} = \frac{0+0.8+0.45}{3} = 0.417 <$ *minDiv*. □

When mining frequent patterns, the frequency/ support measure [2,11] satisfies the downward closure property (i.e., all supersets of an infrequent patterns are infrequent). This helps reduce the search/solution space by pruning infrequent patterns, which in turn speeds up the mining process. However, mining diverse social entities is different from mining frequent patterns, as observed from Example 4 that group G' = {Edyta} is not diverse but its super-group G = {Cyryl, Edyta} is diverse. In other words, diversity does *not* satisfy the downward closure property (i.e., if a group is not diverse, then *not* all of its super-groups are guaranteed to be diverse).

4. Diverse social entity mining: a tree-based strategy

As diversity does *not* satisfy the downward closure property, we cannot prune those groups that are not diverse. Hence, the mining of diverse social entities can be challenging. To handle this challenge, for each domain D, we identify the (*global*) *maximum prominence value* $GMProm_D$ among all friends:

$$GMProm_D = \max_{f_i} Prom_D(f_i). \tag{4}$$

Hence, global maximum prominence values $GMProm_{D_{1,2,\ldots,d}}$ in multiple domains are represented as $GMProm_{D_{1,2,\ldots,d}} = \langle GMProm_{D_1}, GMProm_{D_2}, \ldots, GMProm_{D_d}\rangle$. Then, for each friend group G, we calculate an upper bound of the influence value $Inf^U_D(G)$ by multiplying $GMProm_D$ (instead of the actual $Prom_D(G)$) with the corresponding frequency $Freq_D(G)$:

$$Inf^U_D(G) = GMProm_D \times Freq_D(G) \qquad \geq Inf_D(G). \tag{5}$$

The upper bound of diversity value $Div^U(G)$ can then be computed by using $Inf^U_D(G)$:

$$Div^U(G) = \frac{\sum_{j=1}^{d} Inf^U_{D_j}(G)}{d} \qquad \geq Div(G). \tag{6}$$

Lemma 1. Let G be a group of friends in F_{SN} such that a friend $f_i \in G$. If $Div^U(f_i) < minDiv$, then $Div(G)$ must also be less than $minDiv$. □

Example 5. Let us revisit F_{SN} in Table 2. Global maximum prominence values are $GMProm_{D_1}$=0.90, $GMProm_{D_2}$ = 0.70, and $GMProm_{D_3}$=0.70. Recall from Example 4 that $Freq_{D_{1,2,3}}(\{\text{Edyta}\}) = \langle 0, 2, 1\rangle$. Then, we can compute $Div^U(\{\text{Edyta}\}) = \frac{(0.90\times0)+(0.70\times2)+(0.70\times1)}{3} = 0.7 \geq minDiv$. So, we do not prune {Edyta} to avoid missing its super-group {Cyryl, Edyta}, which is diverse. Similarly, $Div^U(\{\text{Felicja}\}) = \frac{(0.90\times0)+(0.70\times0)+(0.70\times1)}{3}$=0.23 < $minDiv$. Due to Lemma 1, we prune Felicja as none of its super-groups can be diverse. □

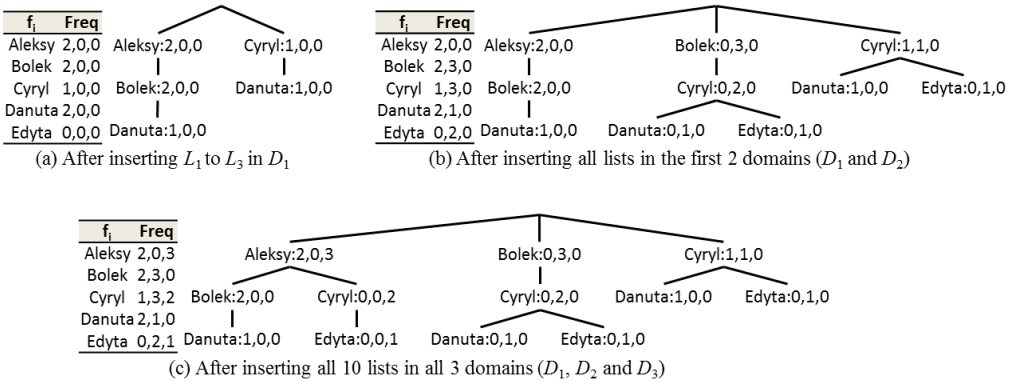
4.1. An overview of our DiSE-growth algorithm

Our proposed DiSE-growth algorithm mines diverse social entities in three phases. In the first phase, the DiSE-growth algorithm takes (i) a friend network F_{SN} and (ii) a user-specified *minDiv* threshold to build a DiSE-tree structure to capture important information about F_{SN}, which include the social entities and their frequencies in each domain in F_{SN}. In the second phase, the DiSE-growth algorithm recursively mines potentially diverse social entities from the DiSE-tree structure built in the first phase. In the third phase, the DiSE-growth algorithms checks all potentially diverse social entities found in the second phase to see if they are truly diverse.

4.2. Our DiSE-growth algorithm builds the DiSE-tree structure

To construct a DiSE-tree, our DiSE-growth algorithm first scans F_{SN} to calculate the frequency $Freq_{D_j}(f_i)$ for each social entity/friend f_i in each domain D_j. The frequency of f_i in domain D_j is bounded above by the number of interest-group lists in each domain D_j. In other words, $0 \leq Freq_{D_j}(f_i) \leq |D_j|$, where $|D_j|$ is the number of interest-group lists in domain D_j.

Then, DiSE-growth scans the table containing the prominence value $Prom_{D_j}(f_i)$ for each social entity/friend f_i in every domain D_j. Based on these prominence values, DiSE-growth computes the global maximum prominence value $GMProm_{D_j}$ for each domain D_j according to Equation (4), i.e., $GMProm_{D_j} = \max_{f_i} Prom_{D_j}(f_i)$. Using these maximum prominence values, DiSE-growth computes the upper bound of the diversity value $Div^U(f_i)$ according to Equation (6), i.e., $Div^U(f_i) = \frac{\sum_{j=1}^{d} Inf^U_{D_j}(f_i)}{d} = \frac{\sum_{j=1}^{d}\left(GMProm_{D_j}\times Freq_{D_j}(f_i)\right)}{d}$. Such an upper bound is then used to prune social entities/friends who are not potentially diverse. In other words, DiSE-growth safely removes any social entity/friend f_i having its $Div^U(f_i)$ below the user-specified *minDiv* threshold because any super-group of f_i cannot be diverse (due to Lemma 1).



(a) After inserting L_1 to L_3 in D_1

(b) After inserting all lists in the first 2 domains (D_1 and D_2)

(c) After inserting all 10 lists in all 3 domains (D_1, D_2 and D_3)

Fig. 1. Construction of a DiSE-tree.

As the remaining social entities/friends are potentially diverse, DiSE-growth stores these f_i—along with their $Freq_{D_{1,\ldots,d}}(f_i)$—in the header table of the DiSE-tree structure.

Afterwards, DiSE-growth scans F_{SN} the second time to capture the important information about potentially diverse social entities in a user-defined order in the DiSE-tree. Specifically, each tree node consists of (i) a friend name and (ii) its frequency counters for all d domains in the respective path. DiSE-growth inserts each interest-group list into the DiSE-tree, in which each tree path keeps only those potentially diverse social entities (i.e., only those in the header table). A newly inserted interest-group list is merged with an existing path (or its prefix containing nodes from the root) of the DiSE-tree only if the same social entities exist in both the newly inserted interest-group list (or its prefix) and the existing path.

An observant reader may notice that this tree construction process is similar to that of the FP-tree[11]. A key difference is that, rather than using only a single frequency counter capturing either the maximum or average frequency for all domains (which may lead to loss of information), we keep d frequency counters capturing the frequency for all d domains. See Example 6.

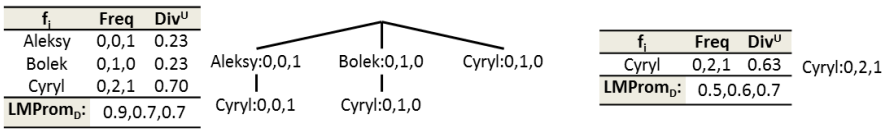
Example 6. To construct a DiSE-tree for F_{SN} shown in Table 2 when *minDiv*=0.5, DiSE-growth scans F_{SN} to compute (i) $GMProm_{D_{1,2,3}} = \langle 0.9, 0.7, 0.7\rangle$ for all d=3 domains, (ii) frequencies of each of the seven friends in d=3 domains (e.g., $Freq_{D_{1,2,3}}(\{\text{Aleksy}\}) = \langle 2, 0, 3\rangle$), (iii) upper bounds of diversity values of all seven friends (e.g., $Div^U(\{\text{Aleksy}\}) = \frac{(0.9\times2)+(0.7\times0)+(0.7\times3)}{3} = 1.3$ using $Inf^U_{D_{1,2,3}}(\{\text{Aleksy}\})$). Based on Lemma 1, we safely remove Felicja and Gustaw having $Div^U(\{\text{Felicja}\})$=0.23 and $Div^U(\{\text{Gustaw}\})$=0.23 both below *minDiv* as their super-groups cannot be diverse. So, the header table includes only the remaining 5 friends—sorted in some order (e.g., lexicographical order of friend names)—with their $Freq_{D_{1,2,3}}(\{f_i\})$. To facilitate a fast tree traversal, like the FP-tree, the DiSE-tree also maintains horizontal node traversal pointers from the header table to nodes of the same f_i.

Our DiSE-growth algorithm then scans each $L_j \in F_{SN}$, removes any friend $f_i \in L_j$ having $Div^U(f_i) < minDiv$, sorts the remaining friends according to the order in the header table, and inserts the sorted list into the DiSE-tree. Each tree node captures (i) f_i representing the group G consisting of all friends from the root to f_i and (ii) its frequencies in each domain $Freq_{D_{1,2,3}}(G)$. For example, the rightmost node "Edyta:0,1,0" of the DiSE-tree in Fig. 1(b) captures G={Cyryl, Edyta} and $Freq_{D_{1,2,3}}(G)=\langle 0, 1, 0\rangle$. Tree paths of common prefix (i.e., same friends) are shared, and their corresponding frequencies are added. See Figs. 1(a), 1(b), and 1(c) for DiSE-trees after reading all interest-group lists in domain D_1, both D_1 and D_2, as well as the entire F_{SN}, respectively. □

With this tree construction process, the size of the DiSE-tree for F_{SN} with a given *minDiv* is observed to be bounded above by $\sum_{L_j \in F_{SN}} |L_j|$.

4.3. Our DiSE-growth algorithm mines all potentially diverse social entity groups

After constructing the DiSE-tree structure, our DiSE-growth algorithm recursively mines/discovers diverse social entity groups by building projected and conditional trees. Specifically, to build a $\{g\}$-projected tree for each social



f_i	Freq	Div^U
Aleksy	0,0,1	0.23
Bolek	0,1,0	0.23
Cyryl	0,2,1	0.70
$LMProm_D$:	0.9,0.7,0.7	

(a) {Edyta}-projected tree

f_i	Freq	Div^U
Cyryl	0,2,1	0.63
$LMProm_D$:	0.5,0.6,0.7	

(b) {Edyta}-conditional tree

Fig. 2. Tree-based mining of diverse social entities.

entity g, DiSE-growth extracts all paths from g to the root. At the end, the projected tree captures all prefixes of interest-group lists containing g.

Recall that $Div(G)$ computed based on $Prom_D(G)$ does *not* satisfy the downward closure property. To facilitate pruning, we use $GMProm_D(f_i)$ to compute $Div^U(f_i)$, which then satisfies the downward closure property. However, if $Div^U(G)$ was computed as an upper bound to super-group G of f_i, then it may overestimate the diversity of G and lead to many false positives. To reduce the number of false positives, DiSE-growth uses the *local maximum prominence value* $LMProm_D(G)$:

$$LMProm_D(G) = \max_{f_i \in F_D^G} Prom_D(G), \tag{7}$$

for the projected tree (which captures F_D^G) for G. See Lemma 2.

Lemma 2. The diversity value of a friend group G computed based on $LMProm_D(G)$ is a tighter upper bound than $Div^U(G)$ computed based on $GMProm_D$:

$$\text{Tightened } Div^U(G) = \frac{\sum_{j=1}^{d}\left(LMProm_{D_j}(G) \times Freq_{D_j}(G)\right)}{d} \qquad \leq \frac{\sum_{j=1}^{d}\left(GMProm_{D_j} \times Freq_{D_j}(G)\right)}{d}, \tag{8}$$

where d is the number of domains in the social network. □

To continue with the mining process, for any $\{g\}$-projected tree built for a social entity g, DiSE-growth computes the diversity value of a friend group $G \equiv (\{g\} \cup \{f_i\})$ based on $LMProm_D(G)$ according to Equation (8). If such a tightened diversity value of G meets or exceeds the user-specified *minDiv*, then G is a potentially diverse social entity group. Otherwise (i.e., the tightened diversity value of G is below *minDiv*), DiSE-growth prunes out G (i.e., removes such an f_i) from the $\{g\}$-projected tree to form a $\{g\}$-conditional tree.

Afterwards, a similar mining process is applied recursively to find other potentially diverse social entity groups. For instance, by extracting appropriate paths from this $\{g\}$-conditional tree, DiSE-growth builds a G-projected tree (for each potentially diverse social entity $G \equiv \{g\} \cup \{f_i\}$), from which (i) a potentially diverse social entity super-group of G can be found and (ii) a G-conditional tree can be formed in preparation for the mining of diverse social entity super-groups. See Example 7.

Example 7. Let us continue with Example 6. To mine potentially diverse social entity groups from the DiSE-tree in Fig. 1(c) using *minDiv* = 0.5, DiSE-growth first builds the {Edyta}-projected tree—as shown in Fig. 2(a)—by extracting the paths (i) ⟨Aleksy, Cyryl, Edyta⟩:0,0,1, (ii) ⟨Bolek, Cyryl, Edyta⟩:0,1,0, and (iii) ⟨Cyryl, Edyta⟩:0,1,0 from the DiSE-tree in Fig. 1(c). For $F^{\text{Edyta}}_{D_{1,2,3}}$ = {Aleksy, Bolek, Cyryl, Edyta}, our DiSE-growth algorithm uses $LMProm_{D_{1,2,3}}(\text{Edyta}) = \langle 0.9, 0.7, 0.7\rangle$ to compute the tightened $Div^U(G)$ such that the tightened $Div^U(\{\text{Aleksy, Edyta}\}) = \frac{(0.9\times0)+(0.7\times0)+(0.7\times1)}{3} = 0.23 < minsup$.

As $Div^U(\{\text{Aleksy, Edyta}\})$ and $Div^U(\{\text{Bolek, Edyta}\})$ are both below *minsup*, DiSE-growth prunes Aleksy and Bolek from the {Edyta}-projected tree to get the {Edyta}-conditional tree as shown in Fig. 2(b). Due to pruning, our DiSE-growth algorithm recomputes (i) the local maximum prominence value $LMProm_{D_{1,2,3}}(\text{Edyta}) = \langle 0.5, 0.6, 0.7\rangle$ and (ii) the tightened $Div^U(\{\text{Cyryl, Edyta}\}) = \frac{(0.5\times0)+(0.6\times2)+(0.7\times1)}{3} = 0.63$ for the updated $F^{\text{Edyta}}_{D_{1,2,3}}$ = {Cyryl, Edyta}. This completes the mining for {Edyta}.

Next, DiSE-growth builds {Danuta}-, {Cyryl}-, and {Bolek}-projected trees as well as their conditional trees, from which potentially diverse social entity groups can be mined. Finally, DiSE-growth computes the true diversity value $Div(G)$ for each of these mined groups to check if it is truly diverse (i.e., to remove all false positives). □

4.4. Our DiSE-growth algorithm removes "false positives" to find all truly diverse social entity groups

Our DiSE-growth algorithm makes good use of the global and local maximum prominence values of friend groups as upper bounds to diversity values of friend groups. Consequently, the algorithm discovers all truly diverse social entity groups (i.e., *no* false negatives). However, it also discover some "potentially diverse" friend groups that are not truly diverse (i.e., some false positives). Hence, as its final step, our DiSE-growth algorithm computes the true diversity values $Div(G)$ for each of these mined groups to check if it is truly diverse (i.e., to remove all false positives). In other words, for each potentially diverse social entity group G having $Div^U(G) \geq minDiv$, DiSE-growth checks and returns G as a truly diverse group if $Div^U(G) \geq Div(G) \geq minDiv$. Those false positive G' having $Div^U(G') \geq minDiv > Div(G')$ are removed.

Example 8. Continue with Example 7. After mining potentially diverse social entity groups from {Edyta}-, {Danuta}-, {Cyryl}-, and {Bolek}-projected trees as well as their conditional trees, our DiSE-growth algorithm computes the true diversity value $Div(G)$ for each of the mined groups to check if it is truly diverse (i.e., to remove all false positives). □

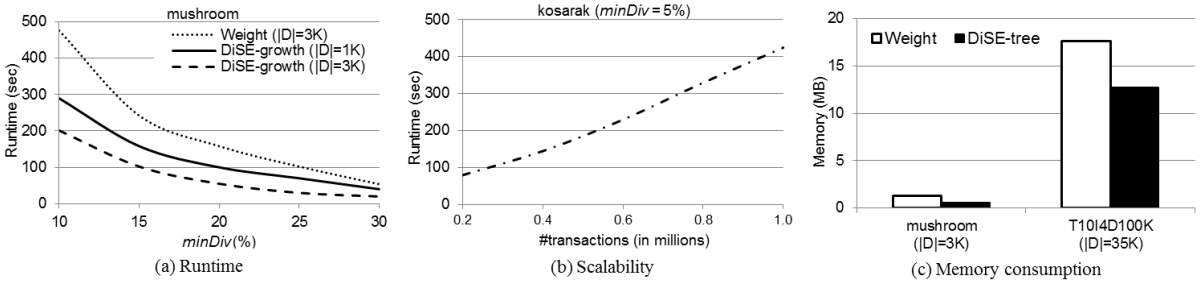
5. Experimental evaluation and analysis

We evaluated the effectiveness of our proposed DiSE-growth algorithm and its associated DiSE-tree structure by comparing them with a closely related *weighted* frequent pattern mining algorithm called Weight[34] (although it does not use different weights for individual items). As Weight was designed for frequent pattern mining (instead of social network mining), we apply those datasets commonly used in frequent pattern mining for a fair comparison: (i) IBM synthetic datasets (e.g., T10I4D100K) and (ii) real datasets (e.g., kosarak, mushroom) from the Frequent Itemset Mining Dataset Repository (http://fimi.ua.ac.be/data). See Table 3 for more detail. Items in transactions in these datasets are mapped into friends in interest-group lists. To reflect the concept of *domains*, we subdivided the datasets into several batches. Moreover, a random number in the range (0, 1] is generated as a prominence value for each friend in every domain. All programs were written in C++ and run on the Windows XP operating system with a 2.13 GHz CPU and 1 GB main memory. The runtime specified indicates the total execution time (i.e., CPU and I/Os). The reported results are based on the average of multiple runs for each case. We obtained consistent results for all of these datasets.

We first compared the runtime of DiSE-growth (which includes the construction of the DiSE-tree, the mining of potentially diverse social entity groups from the DiSE-tree, and the removal of false positives) with that of Weight. Fig. 3(a) shows the results for a dense dataset (mushroom), which were consistent with those for sparse datasets (e.g.,

Table 3. Dataset characteristics.

Dataset	n=#transactions	m=#domain items	Max trans. length	Avg trans. length	Density
kosarak	990,002	41,270	2498	8.1	Sparse
mushroom	8,124	119	23	23.0	Dense
T10I4D100K	100,000	870	29	10.1	Sparse



(a) Runtime (b) Scalability (c) Memory consumption

Fig. 3. Experimental results.

T10I4D100K). Thus, we safely omit the results for sparse datasets. But, runtimes of both algorithms increased when mining larger datasets (social networks), more batches (domains), and/or with lower *minDiv* thresholds. Between the two algorithms, our tree-based DiSE-growth algorithm outperformed the Apriori-based Weight algorithm. Note that, although FP-growth[11] is also a tree-based algorithm, it was *not* design to capture weights. To avoid distraction, we omit experimental results on FP-growth and only show those on Weight (which captures weights).

We then tested the scalability of our DiSE-growth algorithm by varying the number of transactions (interest-group lists). We used the kosarak dataset as it is a huge sparse dataset with a large number of distinct items (individual users). We divided this dataset into five portions, and each portion is subdivided into multiple batches (domains). We set *minDiv*=5% of each portion. Fig. 3(b) shows that, when the size of the dataset increased, the runtime also increased proportionally implying that DiSE-growth is scalable.

We also evaluated the memory consumption. Fig. 3(c) shows the amount of memory required by our DiSE-tree for capturing the content of social networks with the lowest *minDiv* threshold (i.e., without removing any friends who were not diverse). Although this simulated the worst-case scenario for our DiSE-tree, DiSE-tree was observed (i) to consume a reasonable amount of memory and (ii) to require less memory than Weight (because our DiSE-tree is compact due to the prefix sharing).

To summarize, experimental results on (i) runtime, (ii) scalability, and (iii) memory consumption (which reveals tree compactness) showed that our scalable DiSE-growth algorithm is time- and space-efficient. As ongoing work, we plan to measure the *quality* (e.g., precision) of DiSE-growth in finding truly diverse social entity groups. Moreover, for a fair comparison with Weight, we have used those datasets that are commonly used in frequent pattern mining. As ongoing work, we plan to evaluate DiSE-growth using real-life social network datasets.

6. Conclusions and future work

In conclusion, we (i) introduced a new notion of *diverse social entities* for social networks, (ii) proposed a compact tree structure called *DiSE-tree* to capture important information from social networks, and (iii) designed a tree-based mining algorithm called *DiSE-growth* to find diverse (groups of) social entities from social networks. Diversity of friends was measured based on their prominence, frequency and influence in different domains on the networks. Although diversity does not satisfy the downward closure property, we managed to address this issue by using the global and local maximum prominence values of social entity groups as upper bounds. Experimental results showed that (i) our DiSE-tree is compact and space-effective and (ii) our DiSE-growth algorithm is fast and scalable for both sparse and dense datasets. As ongoing work, we are (i) conducting more extensive experimental evaluations with various datasets (e.g., real-life social network datasets) and (ii) measuring other aspects (e.g., precision) of our DiSE-growth algorithm in finding diverse social entities.

As for future work, we plan to (i) design a more sophisticated way to measure influence and (ii) incorporate other computational metrics (e.g., popularity, significance, strength) with prominence into our discovery of useful information from social networks.

Acknowledgements

This project is partially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and the University of Manitoba.

References

1. Agarwal N, Liu H. *Modeling and data mining in blogosphere*. San Hafael, CA: Morgan and Claypool; 2009.
2. Agrawal R, Srikant R. Fast algorithms for mining association rules in large databases. In: *Proceedings of the VLDB 1994*. Morgan Kaufmann; 1994, p. 487–499.
3. Anagnostopoulos A, Kumar R, Mahdian M. Influence and correlation in social networks. In: *Proceedings of the ACM KDD 2008*. ACM; 2008, p. 7–15.
4. Bizer C, Heath T, Berners-Lee T. Linked data - the story so far. *International Journal on Semantic Web and Information Systems* 2009; **5**(3):1–22.

5. Chen YC, Zhu WY, Peng WC, Lee WC, Lee SY. CIM: community-based influence maximization in social networks. *ACM Transactions on Intelligent Systems and Technology* 2014; **5**(2): art. 25.
6. Cuzzocrea A, Folino F. Community evolution detection in time-evolving information networks. In: *Proceedings of the EDBT/ICDT Workshops 2013*. ACM; 2013, p. 93–96.
7. Cuzzocrea A, Folino F, Pizzuti C. *DynamicNet*: an effective and efficient algorithm for supporting community evolution detection in time-evolving information networks. In: *Proceedings of the IDEAS 2013*. ACM; 2013, p. 148–153.
8. Cuzzocrea A, Leung CK, Tanbeer SK. Mining of diverse social entities from linked data. In: *Proceedings of the EDBT/ICDT Workshops 2014*. CEUR-WS.org; 2014, p. 269–274.
9. Dai BT, Kwee AT, Lim EP. ViStruclizer: a structural visualizer for multi-dimensional social networks. In: *Proceedings of the PAKDD 2013, Part I*. Springer; 2013, p. 49–60.
10. Ferrara A, Genta L, Montanelli S. Linked data classification: a feature-based approach. In: *Proceedings of the EDBT/ICDT Workshops 2013*. ACM; 2013, p. 75–82.
11. Han J, Pei J, Yin Y. Mining frequent patterns without candidate generation. In: *Proceedings of the ACM SIGMOD 2000*. ACM; 2000, p. 1–12.
12. Hemsley J, Mason RM. Knowledge and knowledge management in the social media age. *Journal of Organizational Computing and Electronic Commerce* 2013; **23**(1-2):138–167.
13. Jiang F, Leung CK, Tanbeer SK. Finding popular friends in social networks. In: *Proceedings of the CGC* (*SCA*) *2012*. IEEE; 2012, p. 501–508.
14. Kamath KY, Caverlee J, Cheng Z, Sui DZ. Spatial influence vs. community influence: modeling the global spread of social media. In: *Proceedings of the ACM CIKM 2012*. ACM; 2012, p. 962–971.
15. Lee W, Leung CK, Song JJ, Eom CS. A network-flow based influence propagation model for social networks. In: *Proceedings of the CGC* (*SCA*) *2012*. IEEE; 2012, p. 601–608.
16. Leung CK, Carmichael CL. Exploring social networks: a frequent pattern visualization approach. In: *Proceedings of the IEEE SocialCom 2010*. IEEE; 2010, p. 419–424.
17. Leung CK, Carmichael CL, Teh EW. Visual analytics of social networks: mining and visualizing co-authorship networks. In: *Proceedings of the FAC 2011, HCII 2011*. Springer; 2011, p. 335–345.
18. Leung CK, Medina IJM, Tanbeer SK. Analyzing social networks to mine important friends. In: Xu G, Li L, editors. *Social media mining and social network analysis: emerging research*. IGI Global; 2013, p. 90–104.
19. Leung CK, Tanbeer SK, Cameron JJ. Interactive discovery of influential friends from social networks. *Social Network Analysis and Mining* 2014; **4**(1): art. 154.
20. Meng Q, Kennedy PJ. Using field of research codes to discover research groups from co-authorship networks. In: *Proceedings of the IEEE/ACM ASONAM 2012*. IEEE; 2012, p. 289–293.
21. Pernelle N, Saïs F. Classification rule learning for data linking. In: *Proceedings of the EDBT/ICDT Workshops 2012*. ACM; 2012, p. 136–139.
22. Schaal M, O'Donovan J, Smyth B. An analysis of topical proximity in the twitter social graph. In: *Proceedings of the SocInfo 2012*. Springer; 2012, p. 232–245.
23. Su JH, Chang WY, Tseng VS. Personalized music recommendation by mining social media tags. In: *Proceedings of the KES 2013*, Elsevier; 2013, p. 303–312.
24. Talukdera M, Quazia A. The impact of social influence on individuals' adoption of innovation. *Journal of Organizational Computing and Electronic Commerce* 2011; **21**(2):111–135.
25. Tanbeer SK, Jiang F, Leung CK, MacKinnon RK, Medina IJM. Finding groups of friends who are significant across multiple domains in social networks. In: *Proceedings of the CASoN 2013*. IEEE; 2013, p. 21–26.
26. Tanbeer SK, Leung CK, Cameron JJ. Interactive mining of strong friends from social networks and its applications in e-commerce. *Journal of Organizational Computing and Electronic Commerce* 2014; **24**(2–3):157–173.
27. Tang J, Liu H. Unsupervised feature selection for linked social media data. In: *Proceedings of the ACM KDD 2012*. ACM; 2012, p. 904–912.
28. Turban E, Bolloju N, Liang TP. Enterprise social networking: opportunities, adoption, and risk mitigation. *Journal of Organizational Computing and Electronic Commerce* 2011; **21**(3): 202–220.
29. Wu Z, Yin W, Cao J, Xu G, Cuzzocrea A. Community detection in multi-relational social networks. In: *Proceedings of the WISE 2013, Part II*. Springer; 2013, p. 43–56.
30. Xu H, Yang Y, Wang L, Liu W. Node classification in social network via a factor graph model. In: *Proceedings of the PAKDD 2013, Part I*. Springer; 2013, p. 213–224.
31. Yang T, Comar PM, Xu L. Community detection by popularity based models for authored networked data. In: *Proceedings of the IEEE/ACM ASONAM 2013*. ACM; 2013, p. 74–81.
32. Yuan Q, Cong G, Ma Z, Sun A, Magnenat-Thalmann N. Who, where, when and what: discover spatio-temporal topics for twitter users. In: *Proceedings of the ACM KDD 2013*. ACM; 2013, p. 605–613.
33. Zafarani R, Abbasi MA, Liu H. *Social media mining: an introduction*. New York, NY: Cambridge University Press; 2014.
34. Zhang S, Zhang C, Yan X. Post-mining: maintenance of association rules by weighting. *Information Systems* 2003; **28**(7):691–707.
35. Zhong C, Salehi M, Shah S, Cobzarenco M, Sastry N, Cha M. Social bootstrapping: how pinterest and last.fm social communities benefit by borrowing links from facebook. In: *Proceedings of the WWW 2014*. ACM; 2014, p. 305–314.