Finding Contexts of Social Influence in Online Social Networks

Jennifer H. Nguyen[°] Bo Hu[□] Stephan Günnemann[△] Martin Ester[□]

[°]University College London, UK
j.nguyen.12@ucl.ac.uk

[□]Simon Fraser University, Canada
{boh,ester}@cs.sfu.ca

[△]Carnegie Mellon University, USA
sguennem@cs.cmu.edu

ABSTRACT

The ever rising popularity of online social networks has not only attracted much attention from everyday users but also from academic researchers. In particular, research has been done to investigate the effect of social influence on users' actions on items in the network. However, all social influence research in the data-mining field has been done in a context-independent setting, i.e., irrespective of an item's characteristics. It would be interesting to find the specific contexts in which users influence each other in a similar manner. In this way, applications such as recommendation engines can focus on a specific context for making recommendations. In this paper, we pose the problem of finding contexts of social influence where the social influence is similar across all items in the context. We present a full-space clustering algorithm and a subspace clustering algorithm to find these contexts and test the algorithms on the Digg data set. We demonstrate that our algorithms are capable of finding meaningful contexts of influence in addition to rediscovering the predefined categories specific to the Digg news site.

1. INTRODUCTION

Since the introduction of *Web 2.0*, online social networks like Digg have gained in popularity and have become ubiquitous in our every day lives. Just like how these social networking sites are popular among the general public, they are also a popular topic of study among researchers in the data mining field. A particular area of interest is the study of *social influence* in social networks. Social influence theory claims that the behavior of users can influence their network of friends to perform similar actions. The dynamics of a social network is in part governed by this social influence [3, 20, 21, 28].

Previous work has been done to prove the existence of social influence within a social network, as well as maximizing and modeling influence [8, 20, 27]. The presence of social influence can diffuse through a network and can be exploited for tasks like viral marketing, recommendation, and numerous other applications. As such, it is advantageous to learn the varying influence weights of a user on his neighbors. Even more advantageous is to identify the contexts (collection of items) where the influence weights are similar for the items in the context, thereby allowing us to focus on a particular context for viral marketing or other applications [20,23,26]. If a company, for example, wants to sell sports equipment, it should target their advertisements to users showing high influence in the "sport" context. However, all related work in the data mining area has been done in a context-independent setting [8,17,22,27]. In this paper, we investigate social influence in a context-specific setting.

1.1 Motivating Example

Studies in sociology have established that the social influence of an individual varies from context to context [9, 14]. As an example, within the domain of automobiles, user A may have a strong influence on his neighbors because he is an expert in this area. Whereas in the domain of gardening, user A may have little influence because he is not an expert in gardening. Thus, for companies selling automotive equipment, user A might be a good candidate for target marketing, while for companies from different sectors it might be better to select other users.

To illustrate this phenomenon more concretely, we perform a preliminary analysis on the Digg network. Digg is a news-sharing and voting site where users submit news stories and vote on them by "digging". The more votes a story receives, the higher it gains in popularity and is more likely to spread across the network. Six news topics are selected: basketball, celebrities, US 2008 Elections, health, Nintendo, and technology. For each topic, the average influence probability for each user is calculated [8], i.e., the average probability that a user's network of friends voted for a story after he has voted for the same story.

Figure 1 compares the average influence probabilities for the six selected topics and for all topics combined for a random sample of five users. As can be seen, the context-specific influence probabilities vary from topic to topic for each user. For example, user 31 has some influence on his neighbors in the topic of basketball but no influence in the topics of health and Nintendo. We also see that user 122 has a significantly higher influence in all categories except for Nintendo. These results support our hypothesis that the context-specific influence probability of a user is different for different contexts.

Furthermore, when a user's influence probability is measured irrespective of the topic (as in the existing approaches), we lose useful information that can otherwise be obtained if we focus on a certain context. For example, user 31's

The 7th SNA-KDD Workshop '13 (SNA-KDD'13), August 11, 2013, Chicago, United States. Copyright 2013 ACM 978-1-4503-2330-7 ...$5.00.

context-independent influence probability is very small since he has little or no influence in five out of six categories. However, the user obviously has some influence in basketball. This motivates us to find the contexts in which users exhibit significant social influence.

We wish to identify these contexts and the users associated with those contexts who exhibit similar social influence. For example, given the topics in Figure 1, a possible context can be {basketball, election, health and technology} since the influence probabilities of users 50, 53 and 122 are similar across these topics. It would also be interesting to discover novel contexts that are less conventional and not expected such as the grouping of health and Nintendo.

The task of identifying these contexts is a clustering problem. More specifically, it is a graph clustering task as we require the structural properties of the social network graph to measure the social influence probabilities associated with the edges. However, many graph clustering methods perform the clustering based on only the network's structural properties. More recently, there have been new methods that leverage the structure of the network as well as the attributes associated with vertices to find more meaningful clusters [10, 11, 18]. While these methods exploit the attributes on the vertices, we are interested in using the edge attributes (i.e., social influence probabilities) to find clusters.

In this paper we make the following contributions:

- We introduce the novel problem of finding contexts of social influence in a social network.
- Using a social network and social influence probabilities associated with the edges, we develop a full-space clustering model to learn K contexts of social influence within the network.
- We extend the full-space model by presenting a subspace clustering model to also find the corresponding communities of users for each context.
- We carry out experiments to evaluate our models using a real life social network data set and demonstrate the ability of our methods to identify meaningful contexts.

The remainder of the paper is organized as follows: In Section 2, we present a survey of related work. In Section 3, we describe some related concepts and formally introduce the problem definition. Section 4 presents the solution to our problem of interest and Section 5 presents experimental results. We then make some concluding remarks in Section 6.

2. RELATED WORK

In this section, we present a summary of the related work in the area of modeling social influence and clustering in large network graphs.

Social Influence Modeling. For a time, the presence of social influence in online social networks was questionable. After Anagnostopoulos et al. [3] and La Fond et al. [6] developed tests to prove the existence of social influence, the next concern for researchers is how to model this influence and predict it? Rodriguez et al. [21] propose a model to infer the influence propagation given a set of observed user actions and associated timestamps to produce a network that best explains the observed actions times. Xiang et al. [28] develop an unsupervised latent variable model to learn the relationship strengths between users based on user behavior and user similarity, but their model mostly focuses on homophily effects and not social influence.

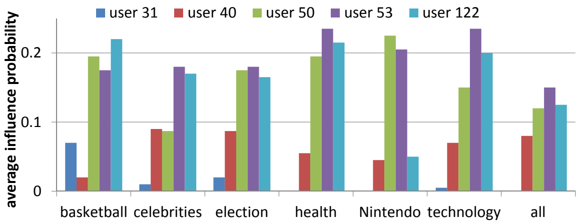


Figure 1: Comparison of influence probabilities for six individual topics and all topics combined

Goyal et al. [8] quantify the influence strengths using influence probabilities. They use the independent cascade model to learn these probabilities, where the influence probability from user u to user v increases with the number of items items v adopts from u. The intuition is that as more of a user's friends perform an action, the more likely the user will perform the same action. Recently, some works [13,19] argue that there are further factors impacting the users' behavior besides social influence. [13] propose a user behavior model, which considers all major factors as social correlation, user, item, and sparsity factors. The work of [22] claims to accurately predict a user's actions, it is also necessary to take into account his passivity. They propose a method to infer the influence scores together with their passivity scores.

All these works, however, do not consider the users' social influence in different contexts as in our paper. As we discussed in the previous section, measuring a user's context-independent influence can be disadvantageous.

Tang et al. [27] introduce the problem of quantifying social influence with respect to different topics and propose a Topical Affinity Propagation approach to model the topic-level social influence in social networks. Liu et al. [17] continue this work by developing a generative graphical model to estimate the direct and indirect topic-based influence between nodes in a network. Their models rely on heterogeneous networks with nodes representing users and items, where each node is associated with a topic distribution. The similarity of their topic distribution is used to infer the (context-independent) degree of social influence between two users. In our paper, we are given a set of topic-specific social influence weights and we are interested in finding the groups of topics that have similar influence weights.

Graph Clustering. To find contexts of social influence, we use a combination of subspace clustering and dense subgraph mining methods [2,15]. Presently, there are relatively few works that combine both methods to take advantage of the different sources of data. Subspace clustering methods use subsets of an object's attributes to find meaningful groups among a collection of objects. We use this approach to find the most useful subset of edges to serve as the support for our contexts. Similarly, dense subgraph mining methods also finds groups of objects (cliques or quasi-cliques) by using the structure of the graph. In our paper, we are require the graph structure and the attributes associated with the edges to find the contexts.

The literature on dense subgraph mining is extensive and we only briefly discuss a few of the relevant works here. In [7], Gibson et al. present a method to find dense subgraphs in a large graph, which employs the shingling algorithm in each recursive step. Another approach by Shi et al. [24] partitions the graph into cuts having small cut size to find dense subgraphs. However, these methods do not incorporate the use of attribute data associated with vertices or edges like in our work.

The works of [12] and [5] introduce a distance function that simultaneously measures the similarity of attributes and network shortest paths, which can then be used with any distance-based clustering algorithm. A limitation of these distance-based methods is that the graph structure is lost in the final output. In [25] a normalized modularity definition where vertex features are incorporated as edge weights is used. For minimizing this normalized modularity, a spectral clustering approach is used. Multiple further clustering methods have been introduced that simultaneously consider network and vector information. Most of the approaches, however, take a full-space clustering perspectives, making their application for high-dimensional data questionable.

The method of [18] combines *subspace clustering* and dense subgraph mining. It aims to find dense, connected subgraphs that are highly similar in a subset of attributes and meet a minimum density threshold. The methods of [10,11] extend this principle by excluding redundant clusters from the final output. While these methods use attribute data associated with the vertices, we introduce a method that uses the attribute data associated with the edges.

3. PROBLEM DEFINITION

In this section, we present some related concepts and formulate our problem definition. For our research problem, we have a social network graph $G = (U, E)$, where $U = \{u_1, u_2, \dots, u_N\}$ is the set of users and $E \subseteq \{e = (u, v) | u, v \in U\}$ is the set of directed edges representing the relationships between users. An edge from u to v indicates that user v is a "follower" or "friend" of user u, henceforth referred to as a *neighbor* of u. In our paper, we assume that the social network is static. That is, all relationships are established at the same time as the creation of the network and no edges are deleted.

In addition to users, there is a set of *items* $I = \{i_1, \dots, i_M\}$, a set of *topics* $T = \{t_1, \dots, t_P\}$ with $t_p \subseteq I$, and a set of *actions* associated with each user.

Definition 1. ***Action:*** *Each user u can perform an action on an item $i \in I$ denoted as $a_u^i = (u, i, d_u^i)$, where d_u^i is the date of the action. The set of all actions performed by a user u is denoted as A_u.*

Please note that each user can perform an action on an item only once. In the Digg data set, the items are news stories and users perform an action by "digging" a story. For simplicity, we assume the set of items is also static. Furthermore, we only consider users who have performed at least one action, i.e., $|A_u| \geq 1$ as they are the most interesting from a social influence perspective. We can infer that an individual is influenced by one of his neighbors if he performs an action on an item after his neighbor performed an action on the same item. Similar to [4], we define an *item adoption* as:

Definition 2. ***Item Adoption:*** *For a pair of neighbors $e = (u, v)$ and an item i, if user u performs an action on item i (i.e., $a_u^i \in A_u$) and v performs an action a_v^i on item i where $d_u^i < d_v^i$, then we define (a_u^i, a_v^i) as an item adoption. Furthermore, we denote the set of all items adopted by v from u as $IA_e = \{i | a_u^i \in A_u \wedge a_v^i \in A_v \wedge d_u^i < d_v^i\}$.*

As noted in the motivating example, users demonstrate different degrees of social influence for different contexts. In this paper, we wish to learn the contexts in which the social influence is similar. We represent a context C as a subset of the topics T, i.e., $C \in 2^T$. For the Digg data set, a context can represent all stories related to the topics basketball and health for example.

To quantify social influence, we define a simple *social influence weight* as in the work of [8]:

Definition 3. ***Social Influence Weight:*** *For every edge $e = (u, v) \in E$ and topic $t \in T$, the social influence weight w_e^t is defined as follows:*

$$w_e^t = \begin{cases} \frac{|IA_e^t|}{|A_u^t|} & \textit{if } |A_u^t| \neq 0 \\ \textit{undefined} & \textit{if } |A_u^t| = 0 \end{cases}$$

where $A_u^t = \{a_u^i \in A_u | i \in t\}$ and $IA_e^t = \{i \in IA_e | i \in t\}$

In the first case, we measure the number of items in topic t the user u has performed an action on, compared to the number of items which have been adopted by v from u. The more items v adopts from u, the more evidence we have that u influences v, resulting in a higher influence weight w_e^t. In particular, if v has not adopted any items, then we assume u has no influence on v and $w_e^t = 0$. In the second case, u has not performed an action on any items in t. Consequently, the edge weight is *undefined* as we cannot infer the degree of social influence from the lack of action by u.

The motivation behind this weighting scheme is to predict a user's future actions. As in the case of viral marketing, an individual's probability of being exposed to a piece of marketing increases as more of his friends are exposed to the same piece of marketing. Similarly, a user in an online social network is more likely to perform an action on an item if many of his neighbors have already performed the same action. This concept is reflected in the work of [8].

We would like to identify subsets of topics that exhibit similar influence patterns in terms of which edges in the network demonstrate high influence and which edges demonstrate low influence for that set of topics. This influence pattern is captured by the topic's *influence signature* represented by an $|E|$-dimensional vector where each element is the social influence weight assigned to each edge in E, i.e.,

$$\boldsymbol{t} = \left(w_{e_1}^t, \dots, w_{e_{|E|}}^t\right) \in \mathbb{R}^{|E|}$$

The topics that form a cohesive context are those that share similar influence signatures because these topics elicit the same behavior from all users in the network. That is, for an ideal context $C \in 2^T$, the same users are being influenced by the same neighbors. Conversely, the users who are not influenced and users who are not influential with respect to a topic in C remain consistent across all other topics in C.

Each context can be characterized by its own influence signature, the *context centroid* $\boldsymbol{\mu}$, that is representative of all $t \in C$.

Definition 4. ***Context Centroid:*** *For a context C, the context centroid is defined as $\boldsymbol{\mu} = \left(\mu_1, \dots, \mu_{|E|}\right)$ where*

$$\mu_e = |W_e|^{-1} \sum_{\substack{t \in C \\ w_e^t \neq undef}} w_e^t, \qquad e = 1, \dots, |E|$$

and $W_e = \{t \in C | w_e^t \neq undef\}$.

Furthermore, to measure the similarity between influence signatures, we define a distance function as follows:

Definition 5. ***Distance Function**: For a pair of influence signatures $\boldsymbol{s}, \boldsymbol{t}$, the distance between these signatures is computed as*

$$Dist(\boldsymbol{s}, \boldsymbol{t}) = |N|^{-1} \sum_{e \in E} \mathit{diff}(w_e^s, w_e^t)$$

where

$$\mathit{diff}(x, y) = \begin{cases} |x - y| & x, y \text{ both defined } \wedge (x \neq 0 \vee y \neq 0) \\ 0 & \text{otherwise} \end{cases}$$

and $N = \{e \in E | w_e^s, w_e^t$ both defined $\wedge (w_e^s \neq 0 \vee w_e^t \neq 0)\}$.

The distance function calculates the difference between two social influence weights of an edge for all edges in the network. However, many users in the network have no influence on any of their neighbors for most topics (i.e, $w_e^t = 0$), resulting in a highly sparse data set. Therefore, if defining the difference as 0 as soon as one of the influence weights is undefined, the distance between two influence signatures $\boldsymbol{s}$ and $\boldsymbol{t}$ will be more or less the same. Thus, we are only concerned with measuring the difference between weights that are both defined and are both not equal to zero. The sum of the differences is normalized by how many edges possess this property so that the distance function does not favor influence vectors with many undefined and 0 influence weights.

With these concepts and definitions established, we present our first problem definition.

Problem 1. ***Finding full network contexts** Given a set of topics T, actions A and a social network graph G, we are interested in partitioning these topics into a set of contexts $\mathcal{C} = \{C_k \in 2^T, k = 1, \ldots, K\}$ (i.e.,$\forall i, j : C_i \cap C_j = \varnothing \wedge \cup_{k=1}^{K} C_k = T$) to minimize*

$$Q_K(\mathcal{C}) = \sum_{k=1}^{K} \sum_{t \in C_k} Dist(\boldsymbol{t}, \boldsymbol{\mu}_k) \tag{1}$$

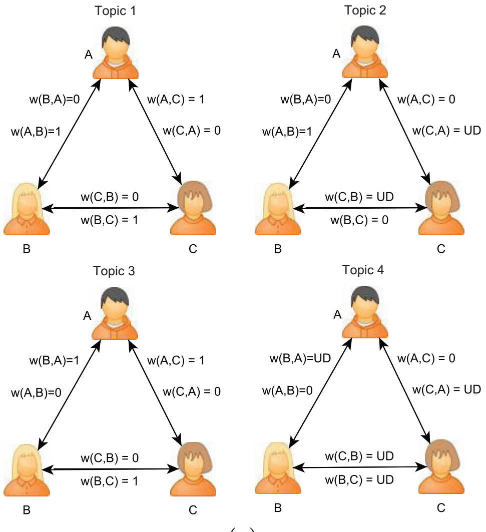
It is possible that even after finding a good set of contexts, some users have little or no influence on their neighbors within those contexts. It makes sense to restrict the context to just those users who exhibit significant influence on their neighbors for topics in the context, i.e., the associated *community* of users for a context. First, we define a partial distance function for a subset of edges as follows.

Definition 6. ***Partial Distance Function**: For a pair of influence signatures $\boldsymbol{s}, \boldsymbol{t}$ and a subset of edges $E' \subseteq E$, the distance between these signatures is computed as*

$$Dist_{E'}(\boldsymbol{s}, \boldsymbol{t}) = |N \cap E'|^{-1} \sum_{e \in N \cap E'} \mathit{diff}(w_e^s, w_e^t)$$

where $\mathit{diff}(x, y)$ and N are defined as in Definition 5.

This partial distance function measures the distance between two influence signatures with respect to a community, i.e., a subset of edges. With this distance function, we ignore those edges that have insignificant influence weights for a context, as they do not contribute any useful information when forming a context. However, since we consider only users u with $|A_u| \geq 1$, every edge has a defined social influence weight for at least one item i and that edge contributes to the formation of at least one context, i.e., the edge belongs to at least one community. Thus, we want to ensure that each edge is associated to at least one context. This leads us to our second problem definition.



(a)

(A, 1, d_1)
(B, 1, d_2)
(C, 1, d_3)
(A, 2, d_1)
(B, 2, d_2)
(A, 3, d_2)
(B, 3, d_1)
(C, 3, d_4)
(A, 4, d_2)

(b)

Topic	Edges					
	(A,B)	(B,A)	(B,C)	(C,B)	(C,A)	(A,C)
1	1	0	1	0	0	1
2	1	0	0	UD	UD	0
3	0	1	1	0	0	1
4	UD	UD	0	UD	UD	0

(c)

	Partitions		
	$C_1 = \{1, 2\}$, $C_2 = \{3, 4\}$	$C_1 = \{1, 3\}$, $C_2 = \{2, 4\}$	$C_1 = \{2, 3\}$, $C_2 = \{1, 4\}$
$Q_2(\mathcal{C})$	1.67	0.5	2

(d)

Figure 2: Toy example demonstrating problem definition. (a) is the social influence graphs for each topic; (b) is the set of actions; (c) is the social influence matrix obtained from the graphs. The blue rows represent context C_1 and the pink rows represent context C_2. The associated edges for each context are highlighted in the corresponding color with yellow edges shared by both contexts. (d) values of $Q_K(\mathcal{C})$ for $K = 2$ contexts with 2 items in each context.

Problem 2. ***Finding subnetwork contexts** Given a set of topics T, actions A and a social network graph G, we are interested in partitioning the topics into K contexts $\mathcal{C}$ as in Problem 1 but to also find a community, i.e., a set of edges $E_k \subseteq E$ for each context C_k with $\cup_{k=1}^{K} E_k = E$ to minimize*

$$Q'_K(\mathcal{C}) = \sum_{k=1}^{K} \sum_{t \in C_k} Dist_{E_k}(\boldsymbol{t}, \boldsymbol{\mu}_k) \tag{2}$$

Note that in the literature a community is often assumed to be connected. However, there may be multiple connected sub-networks that exhibit social influence within the same context but that are not connected to each other. Therefore, our Algorithm 2, presented in Section 4.2, enforces a loose connectivity constraint, i.e. a community is a set of a few connected components.

As an example, suppose we have a network $G = (U, E)$ with $U = \{A, B, C\}$ and $E = \{(A, B), (B, A), (B, C), (C, B),$ $(C, A), (A, C)\}$), a set of items $I = \{1, 2, 3, 4\}$, a set of topics $T = \{t_1, t_2, t_3, t_4\}$, where $t_1 = \{1\}, \ldots, t_4 = \{4\}$, and the actions given in Figure 2b. Using the weighting scheme as defined in Definition 3, each edge in G has a weight with respect to a given topic as shown in Figure 2a. Converting

Algorithm 1 Learning full network contexts

```
Require: $G$, $T = \{\boldsymbol{t}_1, \ldots, \boldsymbol{t}_P\}$, $K$
Ensure: $\mathcal{C} = \{C_1, \ldots, C_K\}$
 1: for all $t \in T$ do
 2:     $C_k \leftarrow t, k \in 1, \ldots, K$ (random assignment)
 3: Initialize: $Q_K(\mathcal{C}) \leftarrow 0$
 4: repeat
 5:     for all $C_k \in \mathcal{C}$ do
 6:         $compute(\boldsymbol{\mu_k})$
 7:     for all $t \in T$ do
 8:         $C_k \leftarrow \underset{k}{\text{argmin}} Dist(\boldsymbol{t}, \boldsymbol{\mu}_k)$, $k \in 1, \ldots, K$
 9:     $previous \leftarrow Q_K(\mathcal{C})$
10:     $update(Q_K(\mathcal{C}))$
11: until $Q_K(\mathcal{C}) \geq previous$
12: return $\mathcal{C}$
```

these graphs to a matrix of influence ratings as in Figure 2c, the optimal partitioning for $K = 2$ contexts is $C_1 = \{1, 3\}$ and $C_2 = \{2, 4\}$ as this produces the most homogeneous contexts and minimizes (1) (cf. Figure 2d).

Furthermore, if we want to restrict the set of edges for each context we can have $E_1 = \{(A, C), (C, A), (B, C), (C, B)\}$ as these are the most similar edges for C_1. For C_2, we first select (B, C) and (A, C) by the same reasoning. To satisfy the constraint that all edges are included in at least one context, we assign the remaining edges to C_2 to get $E_2 = \{(A, B), (B, A), (B, C), (A, C)\}$ and minimize (2) to get $Q'_2(\{C_1, C_2\}) = 0$. If we had assigned the remaining edges to C_1, we get $Q'_2(\{C_1, C_2\}) = 0.5$.

4. ALGORITHMS

In the following section, we present two algorithms to partition the set of topics T into contexts $\mathcal{C} = \{C_1, \ldots, C_K\}$ and minimize the objective functions $Q_K(\mathcal{C})$ and $Q'_K(\mathcal{C})$ in (1) and (2) respectively. The first approach uses the full network G to perform the partitioning while the second approach is an extension of the first and associates a subset of the edges to each context.

Algorithm 1: Full-space clustering.

Minimizing $Q_K(\mathcal{C})$ is equivalent to grouping all $t \in T$ into K homogeneous clusters. Finding the optimal solution for this objective is obviously intractable. Instead, we refer to a near-optimal solution be exploiting a K-means like algorithm that takes into account the data's sparsity. An overview of the method is given in Algorithm 1. The social network graph G, the set of topics T and the number of contexts K are the inputs to the algorithm, which outputs the set of contexts $\mathcal{C}$.

In lines 1-2, the algorithm begins by creating K initial contexts by randomly assigning each item to one of the contexts. In line 6, the context centroid $\boldsymbol{\mu}_k$, $k = 1, \ldots, K$ is computed for each context as defined in Definition 4. In this step, any undefined values are ignored and do not contribute to the calculation of the context centroid. In lines 7-8, each topic is reassigned to the nearest context according to the distance function defined in Definition 5. The reassignments are done to minimize the within-context sum of distances and further minimize $Q_K(\mathcal{C})$ with each iteration. In line 10, $Q_K(\mathcal{C})$ is recomputed as defined in (1) for the current iteration's clustering. These steps are repeated until $Q_K(\mathcal{C})$ has converged to a minimum and no more reassignment of topics occurs.

Since there might exist several partitionings of the topics that locally minimizes $Q_K(\mathcal{C})$ depending on the initial assignment of topics, we perform the algorithm several times and choose the best solution.

Algorithm 2 Learning contexts using subspace clustering

```
Input: $G$, $T = \{\boldsymbol{t}_1, \ldots, \boldsymbol{t}_P\}$, $K$, $L$
Output: $\mathcal{C} = \{C_1, \ldots, C_K\}$
    $\mathcal{E} = \{E_1, \ldots, E_K | E_k \subseteq E\}$ (the set of edges for each context)
 1: for all $t \in T$ do
 2:     $C_k \leftarrow t, k \in 1, \ldots, K$ (random assignment)
 3: Initialize: $Q'_K(\mathcal{C}) \leftarrow 0$
 4: repeat
 5:     for all $C_k \in \mathcal{C}$ do
 6:         $compute(\boldsymbol{\mu_k})$
 7:     for all $C_k \in \mathcal{C}$ do
 8:         $E_k \leftarrow findedges(C_k, L)$
 9:     for all $e \in E \setminus \cup_{k=1}^{K} E_k$ do
10:         $E_k \leftarrow findcommunity(e, \mathcal{C})$, $k \in 1, \ldots, K$
11:     for all $t \in T$ do
12:         $C_k \leftarrow \underset{k}{\text{argmin}} Dist_{E_k}(\boldsymbol{t}, \boldsymbol{\mu_k}, E_k)$, $k \in 1, \ldots, K$
13:     $previous \leftarrow Q'_K(\mathcal{C})$
14:     $update(Q'_K(\mathcal{C}))$
15: until $Q'_K(\mathcal{C}) \geq previous$
16: return $\mathcal{C}$, $\mathcal{E}$
```

Algorithm 2: Subspace clustering.

Besides finding a partitioning of the topics, in our second problem definition, we are interested in finding the supporting communities. Thus, we simultaneously want to find sets of topics and sets of edges. This task corresponds to the principle of subspace clustering [15]. Thus, our second algorithm exploits a subspace clustering approach which is similar to the PROCLUS algorithm [1]. Our method finds the L most relevant subset of edges for each context. The parameter L can be thought of as the minimum size of the context's community and needs to be specified to avoid favoring small communities. Pseudocode for the algorithm is presented in Algorithm 2.

The main difference between Algorithm 2 and Algorithm 1 is the $findedges$ routine (line 8). The routine uses the current clustering to find the most influential community for each context. Specifically, it starts with the most influential edge for a context. For our purposes, the most influential edge for a context C_k is the edge that maximizes the influence weights w_e^t for each $t \in C_k$. Given this edge, the routine grows a community by finding the next most influential edge that is connected to the current community, and adds it to E_k, the subset of edges associated with context C_k.

If the next most influential edge is not connected to the current community, it serves as the starting edge for a new community, thereby allowing a context's community to be composed of one or more connected communities. This is permitted since it likely that there exists disconnected communities that show similar influence patterns for a single context. This process is repeated until L edges are included in the community and is performed for each context. Note that we allow an edge to belong to several contexts so that contexts are not competing for the same edges. In essence, by starting with the most influential edges and growing the communities from these edges, the routine mimics the viral marketing phenomenon where the influence propagation starts with the most influential users. More formally,

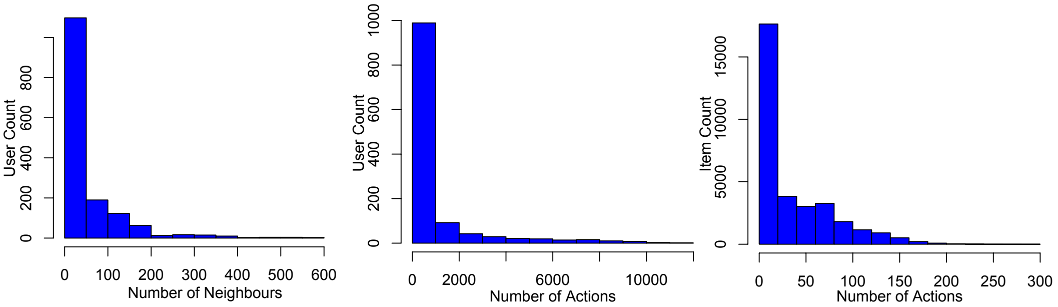
$$findedges(C_k, L) = \underset{\substack{E' \subseteq E, \\ |E'| = L}}{\text{argmax}} \left(\sum_{e \in E'} \sum_{t \in C_k} w_e^t \right)$$

Category	Topics
Lifestyle	arts & culture, autos, educational, food & drink, health, travel & places
Offbeat	comedy, odd stuff, people, pets & animals
Science	environment, general sciences, space
Sports	football, baseball, basketball, extreme sports, golf, hockey, motorsports, Olympics, other sports, soccer, tennis
Technology	Apple, design, gadgets, hardware, Linux/Unix, Microsoft, mods, programming, security, software, tech news
World	business & finance, political news, political opinion, US Elections 2008, world news
Entertainment	celebrity, comics & animation, movies, music, TV
Gaming	gaming, Nintendo, PC games, web games, Playstation, Xbox

Figure 3: Digg categories and topics

users: 1,244
items: 32,496
relations: 33,159
topics: 51
actions: 1,157,529
avg. relations/user: 42.57
avg. actions/user: 930.5
avg. actions/item: 35.6

(a) Statistics



(b) Number of relations per user (c) Number of actions per user (d) Number of actions per item

Figure 4: Characteristics of the Digg social network

Since every edge should participate in at least one context, in lines 9-10 of the algorithm, any unassigned edge is added to the context for which it has the highest influence weights and for which it is connected to that context's community (routine $findcommunity$). If an unassigned edge is not connected to any community, then it is randomly assigned to a context uniformly. Formally, given an edge $e = (u, v)$ and a set of contexts $\mathcal{C}$,

$$findcommunity((u,v),\mathcal{C}) = \begin{cases} \underset{k}{\operatorname{argmax}} \sum_{t \in C_k} w_e^t & \text{if } \exists s \in U | (s,u) \in E_k \vee (s,v) \in E_k \\ uniform(1,K) & \text{otherwise} \end{cases}$$

Note that if $L = |E|$, then Algorithm 2 is the same as Algorithm 1. The remainder of the algorithm is similar to Algorithm 1 but the distances to the context centroid $\boldsymbol{\mu}_k$ are calculated relative to the associated set of edges E_k as in Definition 6.

5. EXPERIMENTS

In this section we test our algorithms on the Digg data set to find contexts of influence and the corresponding community for each context.

5.1 Data

The data used comes from the Digg social network as of 2008 [16]. We pre-process the data to remove any edges (u, v) where neither u nor v has performed any action. We also remove any items from the data where no user has performed an action on it. Figure 4a summarizes the statistics of the Digg social network after the pre-processing step. We see from this table that the Digg dataset is highly connected with an average of 43 neighbors per user, with only a small percentage of users having more than 100 neighbors (cf. Figure 4b). We also see that the network is quite active with 930 actions/user on average. However, the number of actions per user is exponentially distributed as seen in 4c. In terms of the number of actions performed on items, each item has had on average 35 actions performed on it. The number of actions per item is also exponentially distributed, with more than 50% of items being "digged" by only a handful of users (cf. Figure 4d). This pattern is typical of rating networks where there are significantly more items than users.

The news stories (items) in the Digg network are classified into 51 topics (e.g., basketball, baseball, music) and these topics are further classified into eight categories (e.g., basketball and baseball belong to the category of sports) (cf. Figure 3). For each edge e and for each topic t, we compute the social influence weight w_e^t as in Definition 3.

We first run Algorithm 1 to find $K = 4, 6, 8, 10$ contexts of influence. We then apply Algorithm 2 for $L = 400, 8000, 16500, 25000, 33159$ (corresponding to 1%, 25%, 50%, 75%, and 100% of the edges respectively) to possibly improve these contexts and find the corresponding community of users associated with each context.[1]

5.2 Context Analysis

In the absence of ground truth for the contexts, we present a qualitative analysis of the contexts learned by both algorithms. The contexts are analyzed objectively by quality ($Q_K(\mathcal{C}) \backslash Q'_K(\mathcal{C})$) and subjectively by clarity (i.e., how easily interpretable and intuitive are they?). Figure 5 presents the contexts learned by Algorithm 1 for $K = 4, 6, 8, 10$. For $K = 4$, we have one large context (context K4.1) and three smaller contexts. While the poor clarity of context K4.1 does not allow for easy interpretation, the small contexts are a partitioning of predominantly sports, technology and gaming news. These contexts make intuitive sense since most individuals who have an interest in sports typically share an interest in video games, which in turn is closely related to the topic of technology. Furthermore, it is known that 94% of Digg users are male between 18-35 years of age, many of

[1] We implemented our algorithms using the C Clustering Library available at http://bonsai.hgc.jp/~mdehoon/software/cluster/

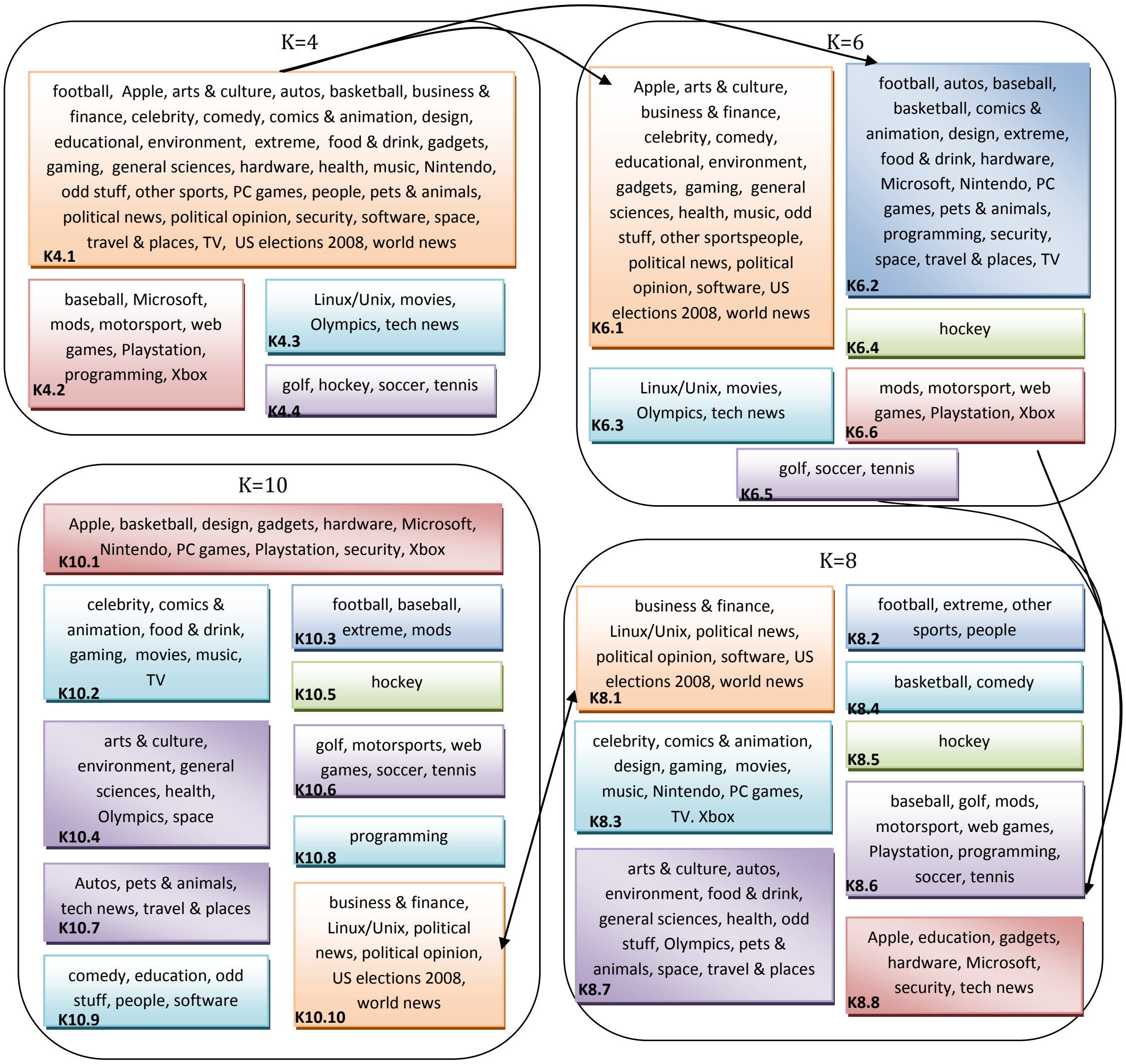

K=4
football, Apple, arts & culture, autos, basketball, business & finance, celebrity, comedy, comics & animation, design, educational, environment, extreme, food & drink, gadgets, gaming, general sciences, hardware, health, music, Nintendo, odd stuff, other sports, PC games, people, pets & animals, political news, political opinion, security, software, space, travel & places, TV, US elections 2008, world news
K4.1
baseball, Microsoft, mods, motorsport, web games, Playstation, programming, Xbox
K4.2
Linux/Unix, movies, Olympics, tech news
K4.3
golf, hockey, soccer, tennis
K4.4
K=6
Apple, arts & culture, business & finance, celebrity, comedy, educational, environment, gadgets, gaming, general sciences, health, music, odd stuff, other sportspeople, political news, political opinion, software, US elections 2008, world news
K6.1
football, autos, baseball, basketball, comics & animation, design, extreme, food & drink, hardware, Microsoft, Nintendo, PC games, pets & animals, programming, security, space, travel & places, TV
K6.2
hockey
K6.4
Linux/Unix, movies, Olympics, tech news
K6.3
mods, motorsport, web games, Playstation, Xbox
K6.6
golf, soccer, tennis
K6.5
K=10
Apple, basketball, design, gadgets, hardware, Microsoft, Nintendo, PC games, Playstation, security, Xbox
K10.1
celebrity, comics & animation, food & drink, gaming, movies, music, TV
K10.2
football, baseball, extreme, mods
K10.3
hockey
K10.5
arts & culture, environment, general sciences, health, Olympics, space
K10.4
golf, motorsports, web games, soccer, tennis
K10.6
programming
K10.8
Autos, pets & animals, tech news, travel & places
K10.7
business & finance, Linux/Unix, political news, political opinion, US elections 2008, world news
K10.10
comedy, education, odd stuff, people, software
K10.9
K=8
business & finance, Linux/Unix, political news, political opinion, software, US elections 2008, world news
K8.1
football, extreme, other sports, people
K8.2
basketball, comedy
K8.4
celebrity, comics & animation, design, gaming, movies, music, Nintendo, PC games, TV. Xbox
K8.3
hockey
K8.5
baseball, golf, mods, motorsport, web games, Playstation, programming, soccer, tennis
K8.6
arts & culture, autos, environment, food & drink, general sciences, health, odd stuff, Olympics, pets & animals, space, travel & places
K8.7
Apple, education, gadgets, hardware, Microsoft, security, tech news
K8.8

Figure 5: Contexts learned by Algorithm 1

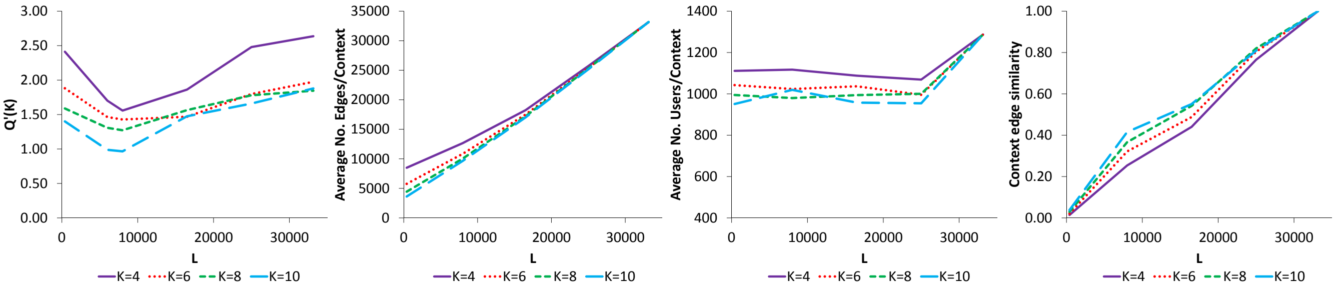
whom work in the technology industry.[2] As such, the majority of actions in the Digg network are performed by this group of users to share and read news in these categories, explaining why these contexts are easily identifiable.

For $K = 6$, we get more clarity in the composition of each contexts. Context K4.1 is now divided into two smaller contexts (K6.1, K6.2). The clarity continues to increase for $K = 8$ and $K = 10$ as contexts become stabilized and the original categories are rediscovered. For example, contexts K8.1/K10.10 and K8.3/K10.2 consists of mostly items from the World category and the Entertainment category respectively. We also see that the Offbeat category is rediscovered in context K10.9. The other contexts are a mixture of sports, technology and gaming news since they are the most popular categories in Digg.

Not only do the learned contexts make sense semantically, but they are also meaningful in terms of social influence. For example, note that hockey news is consistently its own context for $K = 6, 8, 10$ (contexts K6.4, K8.5, and K10.5) and is not grouped with other sports news. After examining the data set, we observe that there are only 30 hockey news stories and they are "digged" by the same group of 200 users. Consequently, this exclusive set of neighbors are influencing each other in a similar manner in only hockey news, resulting in a cohesive context.

For the contexts learned by Algorithm 2, we observe an interesting trend. The quality of the contexts improves as L increases but reaches an optimum at $L = 8000$ (cf. Figure 6a). After this point, the quality of the contexts gets worse. This is because for low values of L, the supporting community is too small and does not provide enough information to form good contexts. The quality of the contexts is best when $L = 8000$ because these edges provide sufficient information to form the best contexts. When $L > 8000$, the contexts become poor again because the algorithm is selecting unnecessary edges (i.e., edges showing influence for very few topics) to associate with each context. These extra edges are not useful and only contribute noise, resulting in a higher value of $Q'_K(\mathcal{C})$. This trend is reflected in the clarity of the contexts.

[2] www.zdnet.com/blog/micro-markets/can-digg-go-mainstream/790



(a) Quality (b) Edges per context (c) Users per context (d) Context edge similarity

Figure 6: Characteristics of learned contexts when using subspace clustering (Algorithm 2)

As seen in Figure 7, the contexts learned for $K = 10, L = 400$ by Algorithm 2 are comparable to the contexts learned by Algorithm 1.[3] However, Figure 7 shows that context L4.1 grows larger, covering a variety of topics as L increases. The contexts stabilizes between $L = 8000$ and $L = 16500$ but then shrinks again as L approaches 33159 and becomes more similar to the contexts learned by the full-space clustering algorithm (cf. Figure 5).

According to Figure 6a, the best quality subspace clustering of the topics is for $L = 8000$ but we see in Figure 7 that context L8.1 has poor clarity. It is large and includes topics from many different categories. A reason for this finding is because the actions in the Digg network are “cheap”. That is, it does not take much risk or commitment to “digg” a news story. Consequently, users are more liberal with the stories they “digg” and tend to perform actions on many stories from a range of topics, resulting in a highly diverse context of topics such as context L8.1.

Ideally, by restricting each community to a subset of the most influential edges, we can get higher quality contexts with better clarity. However, if we refer to Figure 6c, the number of users per context remains relatively constant for all values of L with ∼1000 users per context on average, suggesting nearly the entire network is participating in each context. We also see from Figure 6b that the number of edges per context is always more than L. Even for the case of $K = 6, L = 8000$, which is enough to cover all edges, the algorithm produced contexts with $11,000$ edges on average; much more than expected. This implies that many of the same edges were selected for multiple contexts during the $findedges$ routine in Algorithm 2. Thus, ∼18,000 edges still need to be covered by one of the 6 contexts using the $findcommunity$ routine.

This observation suggests that it is the same 25-35% of neighbors (∼12,000 edges) that are actively interacting with each other across all/most contexts; providing the most useful information to form the best contexts. Furthermore, it explains why the quality of the contexts are poor when $L \geq 16500$. This hypothesis is further confirmed in Figure 6d which shows that the edge similarity between contexts increases as L increases because contexts are selecting the same edges. Yet, the similarity is small because the majority of the edges are the leftover edges that are divided among the contexts. Moreover, the overlap increases as K increases because the same edges are needed by each context. This agrees with the empirical evidence presented in Figure 4a and 4c as only 35% of all users are highly active; having performed greater than the average of 930 actions.

Given the small size of the Digg network, it makes sense to conclude that the entire network of 1244 users form one single community. Thus, by only selecting certain edges for each context, we lose useful information for forming the contexts. These results tell us that to successfully find contexts of influence and the corresponding communities, a large network of users is required and most users of the network need to be actively interacting with each other. It is not sufficient to only having a select group of users perform all the actions. Furthermore, to get contexts with better clarity, actions need to have a higher cost to encourage users to be more conservative when performing actions on items.

6. CONCLUSION

Online social networks have rapidly become a regular part of our everyday lives as a portal to receive and share a variety of media. Several works have proven and quantified the existence of social influence as a factor in guiding users’ actions in these networks. However, the work so far in regards to online social influence has been done in a context-independent setting. In this paper, we introduce the novel problem of finding contexts of similar social influence, where the social influence is uniform across all items in a context. We present a full-space clustering method to find the contexts of social influence and extend this method to only consider certain subsets of edges to find the contexts and the corresponding communities. We test our methods on the Digg data set and show that the full-space method is capable of learning meaningful contexts based on social influence weights. However, due to the small size of the data set, the disproportionate distribution of interactions among users, and the inherent “cheap” actions of the Digg network, the subspace method was not able to find high quality contexts with high clarity. As a direction for future work, we can improve the subspace algorithm to automatically detect the optimal number of edges for each community. In this way, the parameter L does not need to be specified. We also assume that the network is static in this paper, but it would be interesting to investigate how these contexts evolve over time.

Acknowledgement. This research was partially supported by the Natural Sciences and Engineering Research Council of Canada. Stephan Günnemann is supported by a fellowship within the postdoc-program of the German Academic Exchange Service (DAAD).

[3] Since the algorithm consistently produced the highest quality clustering for $K = 10$ (cf. Figure 6a), we only present these contexts in this paper.

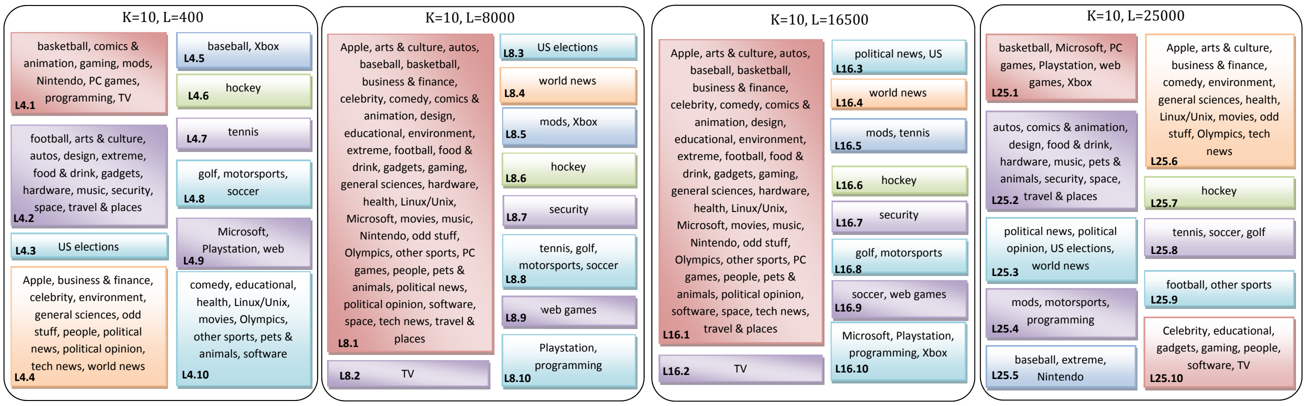

K=10, L=400
basketball, comics & animation, gaming, mods, Nintendo, PC games, programming, TV
L4.1
football, arts & culture, autos, design, extreme, food & drink, gadgets, hardware, music, security, space, travel & places
L4.2
US elections
L4.3
Apple, business & finance, celebrity, environment, general sciences, odd stuff, people, political news, political opinion, tech news, world news
L4.4
baseball, Xbox
L4.5
hockey
L4.6
tennis
L4.7
golf, motorsports, soccer
L4.8
Microsoft, Playstation, web
L4.9
comedy, educational, health, Linux/Unix, movies, Olympics, other sports, pets & animals, software
L4.10
K=10, L=8000
Apple, arts & culture, autos, baseball, basketball, business & finance, celebrity, comedy, comics & animation, design, educational, environment, extreme, football, food & drink, gadgets, gaming, general sciences, hardware, health, Linux/Unix, Microsoft, movies, music, Nintendo, odd stuff, Olympics, other sports, PC games, people, pets & animals, political news, political opinion, software, space, tech news, travel & places
L8.1
TV
L8.2
US elections
L8.3
world news
L8.4
mods, Xbox
L8.5
hockey
L8.6
security
L8.7
tennis, golf, motorsports, soccer
L8.8
web games
L8.9
Playstation, programming
L8.10
K=10, L=16500
Apple, arts & culture, autos, baseball, basketball, business & finance, celebrity, comedy, comics & animation, design, educational, environment, extreme, football, food & drink, gadgets, gaming, general sciences, hardware, health, Linux/Unix, Microsoft, movies, music, Nintendo, odd stuff, Olympics, other sports, PC games, people, pets & animals, political opinion, software, space, tech news, travel & places
L16.1
TV
L16.2
political news, US
L16.3
world news
L16.4
mods, tennis
L16.5
hockey
L16.6
security
L16.7
golf, motorsports
L16.8
soccer, web games
L16.9
Microsoft, Playstation, programming, Xbox
L16.10
K=10, L=25000
basketball, Microsoft, PC games, Playstation, web games, Xbox
L25.1
autos, comics & animation, design, food & drink, hardware, music, pets & animals, security, space, travel & places
L25.2
political news, political opinion, US elections, world news
L25.3
mods, motorsports, programming
L25.4
baseball, extreme, Nintendo
L25.5
Apple, arts & culture, business & finance, comedy, environment, general sciences, health, Linux/Unix, movies, odd stuff, Olympics, tech news
L25.6
hockey
L25.7
tennis, soccer, golf
L25.8
football, other sports
L25.9
Celebrity, educational, gadgets, gaming, people, software, TV
L25.10

Figure 7: Contexts learned by Algorithm 2 for $K = 10$, $L = 400, 8000, 16500, 25000$

7. REFERENCES

[1] C. Aggarwal, C. M. Procopiuc, J. L. Wolf, P. S. Yu, and J. S. Park. Fast algorithms for projected clustering. In *SIGMOD*, pages 61–72, 1999.

[2] C. Aggarwal and H. Wang. *Managing and Mining Graph Data*. Springer, New York, 2010.

[3] A. Anagnostopoulos, R. Kumar, and M. Mahdian. Influence and correlation in social networks. In *KDD*, pages 7–15, 2008.

[4] D. Crandall, D. Cosley, D. Huttenlocher, J. Kleinberg, and S. Suri. Feedback effects between similarity and social influence in online communities. In *KDD*, pages 160–168, 2008.

[5] M. Ester, R. Ge, B. J. Gao, Z. Hu, and B. Ben-Moshe. Joint cluster analysis of attribute data and relationship data: the connected k-center problem. In *SDM*, pages 246–257, 2006.

[6] T. La Fond and J. Neville. Randomization tests for distinguishing social influence and homophily effects. In *WWW*, pages 601–610, 2010.

[7] D. Gibson, R. Kumar, and A. Tomkins. Discovering large dense subgraphs in massive graphs. In *VLDB*, pages 721–732, 2005.

[8] A. Goyal, F. Bonchi, and L. V. Lakshmanan. Learning influence probabilities in social networks. In *WSDM*, pages 241–250, 2010.

[9] M. Granovetter. The strength of weak ties. *American Journal of Sociology*, 78(6):1360–1380, 1973.

[10] S. Günnemann, B. Boden, and T. Seidl. Finding density-based subspace clusters in graphs with feature vectors. *DMKD*, 25(2):243–269, 2012.

[11] S. Günnemann, I. Farber, B. Boden, and T. Seidl. Subspace clustering meets dense subgraph mining: A synthesis of two paradigms. In *ICDM*, pages 845 – 850, 2010.

[12] D. Hanisch, A. Zien, R. Zimmer, and T. Lengauer. Co-clustering of biological networks and gene expression data. *Bioinformatics*, pages 145–154, 2002.

[13] B. Hu, M. Jamali, and M. Ester. Learning the strength of the factors influencing user behavior in online social networks. In *ASONAM*, pages 368–375, 2012.

[14] D. Krackhardt. The strength of strong ties: The importance of philos in networks and organization. In *Networks and Organizations*. 1992.

[15] H. P. Kriegel, P. Kroger, and A. Zimek. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *TKDD*, 3(1), 2009.

[16] Y. R. Lin, J. Sun, P. Castro, R. Konuru, H. Sundaram, and A. Kelliher. Metafac: Community discovery via relational hypergraph factorization. In *KDD*, pages 527–535, 2009.

[17] L. Liu, J. Tang, J. Han, M. Jiang, and S. Yang. Mining topic-level influence in heterogeneous networks. In *CIKM*, pages 199–208, 2010.

[18] F. Moser, R. Colak, A. Rafiey, and M. Ester. Mining cohesive patterns from graphs with feature vectors. In *SDM*, pages 593–604, 2009.

[19] S. A. Myers, C. Zhu, and J. Leskovec. Information diffusion and external influence in networks. In *KDD*, pages 33–41, 2012.

[20] M. Richardson and P. Domingos. Mining knowledge-sharing sites for viral marketing. In *KDD*, pages 61–70, 2002.

[21] M. Gomez Rodriguez, J. Leskovec, and A. Krause. Inferring networks of diffusion and influence. In *KDD*, pages 1019–1028, 2010.

[22] D. M. Romero, W. Galuba, S. Asur, and B. A. Huberman. Influence and passivity in social media. In *WWW*, pages 113–114, 2011.

[23] J. J. Samper, P. A. Castillo, L. Araujo, and J. J. M. Guervós. Nectarss, an rss feed ranking system that implicitly learns user preferences. *CoRR*, abs/cs/0610019, 2006.

[24] J. Shi and J. Malik. Normalized cuts and image segmentation. 22(8):888–905, 2000.

[25] M. Shiga, I. Takigawa, and H. Mamitsuka. A spectral clustering approach to optimally combining numerical vectors with a modular network. In *SIGKDD*, pages 647–656, 2007.

[26] X. Song, Y. Chi, K. Hino, and B. L. Tseng. Information flow modelling based on diffusion rate for prediction and ranking. In *WWW*, pages 191–200, 2007.

[27] J. Tang, J. Sun, C. Wang, and Z. Yang. Social influence analysis in large-scale networks. In *KDD*, pages 807–816, 2009.

[28] R. Xiang, J. Neville, and M. Rogati. Modeling relationship strength in online social networks. In *WWW*, pages 981–990, 2010.