Automatic Social Circle Detection Using Multi-View Clustering

Yuhao Yang, Chao Lan, Xiaoli Li, Bo Luo, and Jun Huan
Department of Electrical Engineering and Computer Science
The University of Kansas, Lawrence, KS 66045, USA
{yyang, clan, xiaolili, bluo, jhuan}@ittc.ku.edu

ABSTRACT

With the development of information technology, online social networks grow dramatically. They now play a significant role in people's social life, especially for the younger generation. While huge amount of information is available in online social networks, privacy concerns arise. Among various privacy protection proposals, the notions of *privacy as control* and *information boundary* have been introduced. Commercial social networking sites have adopted the concept to implement mechanisms such as Google circles and Facebook custom lists. However, the functions are not widely accepted by the users, partly because it is tedious and labor-intensive to manually assign friends into circles.

In this paper, we introduce a social circle discovery approach using multi-view clustering. First, we present our observations on the key features of social circles: friendship links, content similarity and social interactions. We propose a one-side co-trained spectral clustering algorithm, which is tailored for the sparse nature of social network data. We also propose two evaluation measurements. One is based on quantitative similarity measures, while the other employs human evaluators to examine pairs of users selected by the max-risk evaluation approach. We evaluate our approach on ego networks of twitter users, and compare the proposed technique with single-view clustering and original co-trained spectral clustering techniques. Results show that multi-view clustering is more accurate for social circle detection; and our proposed approach gains significantly higher similarity ratio than the original multi-view clustering approach.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Clustering*

Keywords

Social Circles, Social Network, Privacy, Multi-View Clustering

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
CIKM'14, November 3–7, 2014, Shanghai, China.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-2598-1/14/11 ...$15.00.
http://dx.doi.org/10.1145/2661829.2661973.

1. INTRODUCTION

Online social networks are becoming extremely popular, attracting huge amounts of users and Internet traffic. For instance, Facebook recorded one billion active user accounts in late 2012, while approximately 10 million messages are posted every hour. They have significantly changed our information sharing and socialization behavior, especially among the younger generation – it has been reported that 48% percent of Facebook users between 18-34 years old check Facebook when they wake up[1].

The extreme popularity of online social networks has become a double-edged sword. While service providers devote to promote online socialization, privacy issues arise. In the literature, studies have shown a massive disconnection between users' privacy perceptions and their behaviors – widely known as the *privacy paradox*. That is, most users do not take appropriate actions to protect their information, although they express concerns on the privacy of such information [24, 3, 34]. For instance, many users are concerned about their location privacy [20, 7], however, a blog/micro-blog post about a local restaurant [25], or blogs with location-indicating words such as "Time Square" [10, 8] could effectively reveal the user's location. The user-centered privacy and HCI research community has introduced the notion of *restricted access and limited control* [11, 42] and *information boundaries* [38]. In particular, social circles have been proposed for privacy protection [40, 41], so that new messages are posted to designated social circles and the message owners have full control of the information boundary. Meanwhile, social circles are also expected to promote information sharing, since they give users the perception of security and privacy. Various products have been released by commercial social networking sites, such as circles in Google+ and custom lists in Facebook. However, none of them is well-received by users. A major drawback is the usability problem – it is tedious and labor-intensive to assign hundreds of existing friends into circles or lists.

The problem of social community discovery has been studied in the context of social network evolution. Closely-related social groups are examined to analyze the temporal and spatial dynamics of social networks. However, such approaches heavily rely on structural features (i.e., topology of the friendship graph), and may have difficulties on users with too many or too few links (sometimes referred-to as "hubs" and "outliers"). Meanwhile, social circle identification approaches from the user-centered research community often require explicit attributes, e.g. education="stanford",

[1]http://www.statisticbrain.com/facebook-statistics/

age=21, hobbies="hiking" [40]. Unfortunately, such attributes are not always available in online social networking sites.

In this paper, we present a multi-view clustering approach to automatically discover social circles in users' ego networks. Besides the topology-based clusters adopted in the literature, we also observe that: (1) friends who are interested in similar topics (contents) and share similar (or sometimes opposite) opinions are more likely to be placed in the same circle (by the user); (2) friends are more likely to interact within circles, than cross circles. Based on the observations, we build computational models to extract multiple quantitative features from users' ego networks. We argue that integrating all structural, content and interaction features will improve clustering performance, and eventually generate more meaningful social circles. We notice that some views are very sparse (e.g. the views for user interactions), but they provide stronger indications, when two friends are associated in such sparse views. To better utilize such properties, we present a *Selective Co-Trained Spectral Clustering* (SCSC) algorithm for multi-view clustering. Last, to measure the performance of the proposed modeling and clustering approaches, we introduce a set of quantitative and user-based evaluation methods. We test our approaches with real-world social networking data collected from Twitter, and show that SCSC outperforms existing solutions.

Our contributions are three-fold: (1) we are the first to integrate structural, content and interaction features to identify social circles in online social networks; (2) we introduce a novel selective co-trained spectral clustering method to better handle view inconsistency and view sparsity; and (3) we implement and evaluate our methods against real-world social networking data, and demonstrate the superior performance of the proposed approaches.

The rest of the paper is organized as follows: we first present our models of three categories of features in Section 2. We then describe the multi-view clustering algorithms in Section 3. We present our experimental results in Section 4. We further discuss some important issues and our future work in Section 5, provide a brief survey of the literature in Section 6, and conclude the paper in Section 7.

2. EGO NETWORK MODELING

By definition, a user's ego network or personal network includes all the nodes that connect to the user, i.e., all his/her friends. Social circles of a user's ego network are hidden structures of closely connected clusters, For instance, a user's high-school friends may constitute a circle, while his/her colleagues belong to a different circle, and his/her family members constitute yet another circle.

Existing research on social community discovery mostly rely on graph topology, i.e., structural features. However, social circles may not be revealed by a single aspect of the ego networks. Instead, they need to be inferred from multiple features. For example, colleagues may interact frequently offline so that they have few online interactions, however, they are highly *connected* to each other in the friendship graph. Meanwhile, a family member may be connected to some close colleagues on the friendship graph, however, he/she will mostly *interact* with other family members, which is a definitive indicator that he/she belongs to the family circle.

Example 1: Figure 1 demonstrates a small subgraph crawled from Twitter. Two users are regarded as friends if they mutually follow each other. The subgraph is extracted from friends of one seed user. For simplicity, the seed user is not displayed. First, Figure 1 (a) demonstrates the friendship graph – solid lines indicate direct friendship relations, while dashed lines indicate users without direct connections but have shared friends. All the lines are labeled with the number of shared friends (excluding the seed). Next, Figure 1 (b) summarizes the interactions among the users. Each edge is labeled with (N_{rp}, N_{rt}), which indicates the number of replies and re-tweets between the two users, respectively. Last, Figure 1 (c) demonstrates the content similarities between each pair of users (only labels ≥ 0.0065 are shown). We show edges with labels ≥ 0.01 in thick lines.

As shown in the graphs, three views confirm and complement each other in different regions. For instance, the strong connection between nodes A and B in Figure 1 (a) is confirmed by their frequent interactions in Figure 1 (b). The weak connection between C and E in Figure 1 (a) could be eliminated given the facts in Figure 1 (b) and (c). Nodes G and F are disconnected in Figure 1 (a), however, they have a large amount of interactions and very high similarities in their tweet contents, which also indicates a close relationship. In summary, we identify three social circles from this example: $\{(A, B, C); (D, E); (G, F)\}$. As we can see, different perspectives can supplement and confirm each other, which may be utlized to produce better clustering results.□

Formally, an *ego network* E_S is defined as the subgraph of the social network that includes all the friends of a *seed user* S. Note that the seed user himself is not included in the ego network. In the Twitter data set we used, two users are defined as *friends* if and only if they follow each other. In the ego network, each vertex (N_i) represents a friend of the seed user, while the edges are defined differently for different views. In general, we have observed three phenomena about users' grouping behavior:

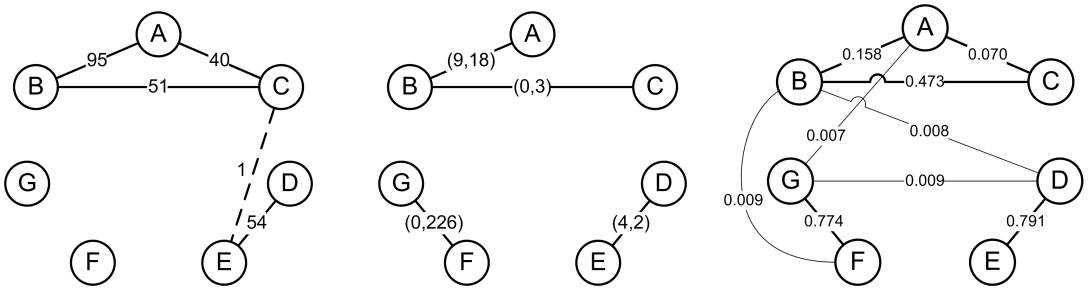
Observation 1. *Users in the same circle are more likely to be connected and share many friends in common.*

Observation 2. *Users from the same social circle tend to share interests on similar contents and opinions.*

Observation 3. *Users in the same circle are more likely to interact with each other.*

From these observations, we propose to integrate three aspects of information from users' ego networks to automatically identify non-overlapping social circles. We define six views that belong to three categories to model the ego networks. From the *structural* perspective, we capture: (1) the friendship links, and (2) friends-in-common between pairs of users. From the *content* perspective, we model: (3) similarities between two users' posted/shared messages. Finally, from the *interaction* perspective, we construct: (4) direct replies between pairs of users, (5) re-tweet (similar to "forward") of posts between pairs of users, and (6) co-replies of the same message (posted by a third user).

The Structural Model. In social networking research, it is widely accepted that a group of intensively connected nodes could be considered as a social community. For each pair of users, they are more structurally connected if they (1) are friends and/or (2) share more friends. We quantitatively capture the structural features in these two layers and create two views correspondingly.



(a) Label: Common Friends (b) Label: Interaction (C) Label: Content Similarity

Figure 1: Labeled Real World Online Social Network Subnet

We use an adjacency matrix F to capture the first layer.

$$F(i,j) = \begin{cases} 1 & \text{if } N_i \text{ and } N_j \text{ are friends.} \\ 0 & \text{if } N_i \text{ and } N_j \text{ are not friends} \end{cases}$$

Meanwhile, the matrix of shared friends (H') for an ego network E_S is defined as:

$$H'(i,j) = |E_{N_i} \cap E_{N_j}| - 1$$

where E_{N_i} and E_{N_j} denote the ego networks of users N_i and N_j. Note that, we consider shared friends within and outside of the original ego network E_S. We do not count S as a shared friend, since S contributes equally to all (N_i, N_j) pairs. Furthermore, the matrix is normalized by dividing each element by the largest element in the matrix:

$$H(i,j) = \frac{H'(i,j)}{\max_{i,j} H'(i,j)}$$

Eventually, we have generated two views F and H to capture the structural relationships between pairs of users in E_S.

The Content Model. From the content perspective, we examine the semantic similarities of contents between pairs of users in E_S. We collect all the tweets, replies and re-tweet messages posted by a user. We exploit the traditional bag-of-words model, where all the messages posted by the user are represented as a vector ($\mathbf{D_i}$) in the vector space. While the conventional TF-IDF model is the most popular method in information retrieval applications, it suffers from some drawbacks, especially the ambiguity issue – synonyms are considered orthogonal axes in the term space. Hence, documents about similar content but from different vocabularies will be assessed as highly irrelevant. To tackle this problem, annotation-based approaches have been proposed to label documents with pre-selected unambiguous terms (topics) so that documents are represented in the new unambiguous "topic space". In this paper, we employ TagMe [12], which annotates text corpus with topics in Wikipedia. Each tag is associated with a "goodness" score, ρ, which denotes the annotating confidence. By setting a threshold for ρ, we can eliminate all the low-confident tags to reduce noise and ambiguity, and improve the calculation efficiency. In practice, we construct a document vector $\mathbf{T_i}$ for user N_i, where each component represents the corresponding TF-IDF weight in the tag space. The content-based similarity matrix C', with cosine similarity, is further defined as:

$$C'(i,j) = sim(\mathbf{T_i}, \mathbf{T_j}) = \frac{\mathbf{T_i} \cdot \mathbf{T_j}}{|\mathbf{T_i}||\mathbf{T_j}|}$$

We normalize C' in the same way as we normalize the shared-friend matrix (H'). Finally, we have constructed the content view for ego network E_S, to capture the content-based similarities among the users.

The Interaction Model. Interactions of online social network users have different forms: reply on each other's status or posted messages, "like" or "dislike" on the messages, retweet. etc. For each pair of nodes within an ego network, we consider three types of interactions: reply, retweet, and co-reply. For reply, we count both directions – the total number of replies from N_i to N_j and replies from N_j to N_i. Therefore, the reply matrix could be denoted as:

$$P(i,j) = |\{\vec{r}_{i,j}\}| + |\{\vec{r}_{j,i}\}|$$

We do the same for retweet, while co-reply is undirected. In this way, we generate three views, and normalize them as we do with H' and C'. As a result, we have constructed the reply view P, the re-tweet view T, and Co-reply view O.

Overall, we construct six views from personal networks: two from the structural perspective, one for content, and the other three from user interactions. Each view is represented as a matrix demonstrating similarities between each pair of users within an ego network. The next step is to integrate these views to identify social circles.

3. MULTI-VIEW CLUSTERING

3.1 Notations and Operators

We use capital letter to represent matrix, boldface to represent vector and lower-case to represent scaler. Subscript without parenthesis is used to indicate views, subscript with parenthesis is used to indicate elements in matrices or vectors, and superscript is used to indicate iteration number in an iterative algorithm. For example, $X_{(m,n)}$ represents the element of matrix X on row m and column n, and X_j^i represents matrix X of view j in the i_{th} iteration.$tr(S)$ is used to denote the trace of S matrix, $A \circ B$ to denote the Hadamard product (element-wise product) between matrix A and matrix B, and 1_E to denote the element-wise indicator function on E. For convenience we define two operators in Operator 1 and Operator 2.

3.2 Co-trained Spectral Clustering: A Revisit

In this section we briefly review co-trained spectral clustering (CSC) [21], which is a clustering algorithm for multi-view data. In spectral clustering, it has been shown that the eigenvectors of the graph Laplacian contains robust dis-

Operator 1: $LapEig(X, k) = Y$

Input: $X \in \mathcal{R}^{n\times n}$ and $k \in \mathcal{N}$
Output: $Y \in \mathcal{R}^{n\times k}$
Operation:
1: Compute diagonal matrix D with $D_{(ii)} = \sum_{j=1}^{n} X_{(ij)}$
2: Compute Laplacian $L = D^{-1/2} X D^{-1/2}$
3: Compute the top k eigenvectors of L, and store them in Y with each column as one eigenvector

Operator 2: $Cls(X, k) = Y$

Input: $X \in \mathcal{R}^{n\times k}$ and $k \in \mathcal{N}$
Output: $Y \in \mathcal{R}^{n\times n}$
Operation:
1: Normalize each row of X
2: Run k-means on rows of X to obtain an n by n matrix Y such that $Y_{(i,j)} = 1$ if user i and user j are in the same cluster, and $Y_{(i,j)} = -1$ if the two users are not in the same cluster

criminative information about the cluster; hence by applying standard clustering techniques on the eigenvectors may lead to a better clustering result. When multiple views of data are available, CSC alternately refines the graph Laplacian of one view based on the clustering result suggested by other views. The refinement is realized by projecting and reconstructing the Laplacian of one view onto the eigenvectors of the graph Laplacians of other views. This process iterates and glues the graph edges within a cluster and differs edges between clusters. The final clustering result is obtained by performing single-view spectral clustering on the refined Laplacians of dominant views.

CSC assumes the graph of each view is completely observed, and transfers the complete graph information across views. However, in our application, graphs of many views are partially observed (hence extremely sparse). For example, intimate users may communicate frequently by replying to each other, while ignoring retweeting messages. In this scenario, the nonexistence of the link between these two nodes in the retweet view should not be used to push these two nodes apart. Similarly, while interaction between two users is a high indication of their intimacy, it would be too permissive to advocate a negative relationship between them if no interaction is observed in retweet or reply views. That is, we hypothesize that enforcing a completely agreement between the sparse views and other views will mis-refine Laplacians and degenerate clustering performance. As we have justified in experimental study, this is indeed a problem.

3.3 Selective Co-trained Spectral Clustering

In this section we propose the new multi-view clustering algorithm. We identify a graph as partial if the number of non-zero entries (edges) in the graph Laplacian is below a pre-specified threshold, and safely assume that only non-zero edges in partial graphs are observed. Our intuition is that, only clustering results on observed edges should be transferred from views with partial graphs.

The proposed Selective Co-trained Spectral Clustering (SCSC) multi-view clustering approach is presented in Algorithm 1. It is different from CSC in two aspects: 1) CSC uses eigen-

Algorithm 1 Selective Co-trained Spectral

Input: Similarity matrix of two views: K_1, K_2
Output: Cluster matrix C
Initialize: $U_j^0 = LapEig(S_j^i, k)$, $C_j^0 = Cls(U_j^0, k)$, $j \in I_v$
$K_{all} = \{K_j\}_{j\in I_v}$, $C_{all}^0 = \{C_j^0\}_{j\in I_v}$
for $i = 1$ **to** *iter* **do**
  for $j = 1$ **to** *views* **do**
    1: $R_j^i = SO(C_{all}^{i-1}, K_{all}, j)$
    2: $S_j = R_j^i \circ K_j$
    3: $U_j^i = LapEig(S_j, k)$
    4: $C_j^i = Cls(U_j^i, k)$
  end for
  5: $C_{all}^{i-1} = \{C_j^{i-1}\}_{j\in I_v}$
end for
6: Choose the dominant view j and run $Cls(U_j^i, k)$ to get the cluster matrix.

vectors of the Laplacian of one view to refine Laplacians of other views, while SCSC uses clustering result of one Laplacian to refine other Laplacians, which tends to be more precise; 2) CSC transfers the complete graph information across views, while SCSC selectively transfers graph information.

Operation $SO(\cdot)$ realizes the selective process. Let ρ_j be defined as

$$\rho_j = \frac{\#\text{ zeros in } K_j}{\#\text{ all elements in } K_j}, \tag{1}$$

where K_j represents the similarity matrix of view j, and abbreviate $SO(C_{all}^{i-1}, K_{all}, j)$ as $SO(j)$, we design

$$SO(j) = \exp\left(C_{j'} \circ \left(1_{\{K_{j'}\neq 0\}}\right)^{1\{\rho_{j'} > \rho_{thre}\}}\right), \tag{2}$$

where $C_{j'}$ represents the clustering matrix of view j' such that if user p and user q are assigned to the same cluster in this view, then the p_{th} row and q_{th} column of $C_{j'}$ is 1, otherwise it is zero. In addition, $j' \in I_v, j' \neq j$ and ρ_{thre} is a pre-specified threshold. The intuitions behind $SO(j)$ are as follows:

- For $C_{j'}$, if two users are assigned to the same cluster in view j', then the corresponding element in $C_{j'}$ is 1 and $SO > 1$ so their Laplacian in other views will be boosted, and vice versa.
- For $I_{K_{j'}\neq 0}$, if two users have non-zero edges in the partial graph of view j', then $I = 1$ and their clustering result in view j' will be transferred to other views; otherwise $SO = 1$ and their Laplacian in other views will not be affected.
- $I_{\{\rho_{j'} > \rho_{thre}\}}$ is used to identify the views that have partial graphs based on threshold ρ_{thre}. Given a view with partial graph, we have $I = 1$ and hence the selective component $I_{\{K_{j'}\neq 0\}}$ will take effect; otherwise $I = 0$ and all information of the graph is transferred.

It is worth elaborating more on the convergence property of our algorithm. Consider an example of clustering three users with two views. From one view we obtain the clustering result

$$C_1 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 1 & -1 & 1 \end{bmatrix}, \tag{3}$$

which implies user 1 and user 3 are similar while user 2 ad user 3 are not. For view two we have the similarity matrix (not the Laplacian)

$$K_2 = \begin{bmatrix} 1 & 0.8 & 0.1 \\ 0.8 & 1 & 0.6 \\ 0.1 & 0.6 & 1 \end{bmatrix}, \quad (4)$$

which implies user 2 and user 3 are similar while user 1 and user 3 are not. Apparently there is certain consistency between views. After refining K_2 by C_1 based on Algorithm 1, we have the updated similarity K_2' such that

$$K_2 = \begin{bmatrix} 2.7 & 2.2 & 0.3 \\ 2.2 & 2.7 & 0.2 \\ 0.3 & 0.2 & 2.7 \end{bmatrix}. \quad (5)$$

It can be seen that after refinement, we manage to adjust user 2 and 3 in view two, so that their updated (low) similarity becomes consistent with the clustering result of view one. However, for user 1 and 3, due to their strong evidence of dissimilarity in view two, they remain dissimilar after refinement.

The above example shows that, our algorithm can effectively adjust and converge on users whose similarity are "uncertain" in some views, but does not enforce agreement and converge on users with strong evidence on two views that are against each other. This is the same issue with existing co-trained spectral clustering, as evidenced by its empirical performance on real-world data sets that contain view-inconsistent (noisy) data. Moreover, in a view with partial graph, we may have massive strong evidence of dissimilarity between users, which is largely against their similarities in other views. In this case, our selective algorithm is expected to gain much faster convergence rate than non-selective clustering algorithms, as will be seen in our experimental study. In addition, notice that since update matrix C is symmetric, the refined similarity matrix K remains symmetric and corresponding Laplacian remains positive semi-definite, which is guaranteed to have real positive eigenvalue.

To extend $SCSC$ to multi-view setting, we let $j = \{1, 2, ..., \ell\}$ and define $SO(j)$ as:

$$SO(j) = \exp\left(\sum_{\substack{j' \neq j \\ j'=1}}^{\ell} C_{j'} \circ \left(I_{\{K_{j'} \neq 0\}} \right)^{I_{\{\rho_{j'} > \rho_{thre}\}}} \right). \quad (6)$$

The summation in (6) follows the majority voting principle: if two users are grouped in more than half of the other views, then their similarity in the current view should be boosted; otherwise their similarity should be decreased. More interestingly, for C_j' if half of the views group two users while the other half separate them, then the summation equals zero and $SO(j) = 1$. In this case, we maintain the similarity of the current view.

4. EXPERIMENT RESULTS

4.1 Evaluation Metrics

To evaluate the quality of our clustering result is quite challenging since no ground truth is provided and no feature matrix is available (in fact, not even defined) in most views. This prevents the use of standard external evaluation metrics such as random index or F-measure or internal evaluation metrics such as Davies-Bouldin index and Dunn index. Hence we propose a new internal metric that requires only the similarity matrix. We believe that better clustering should group users that are not only structurally cohesive (more friendship relations among them), but also interact more frequently and post similar content. Based on this, our evaluation is tripartite.

We first propose the normalized similarity ratio to evaluate the performance of clustering result for each view. Our design follows the same idea as Fisher ratio [4], and consists of three parts, i.e., within-cluster similarity, between-cluster similarity and sparse degree. To prevent the result from being dominated by extra-large clusters, we normalize each similarity by the size of assigned clusters. Consider an arbitrary view, let d_i denote the size of the cluster (number of users in that cluster) assigned to user i, and recall that K is the similarity matrix and C is the cluster matrix. We define the within-cluster similarity as

$$S_{wc} = \frac{1}{N_w} \sum_{i=1}^{\ell} \sum_{j=1}^{\ell} K_{(i,j)} I_{\{C_{(i,j)} > 0\}} \quad (7)$$

and the between-cluster similarity as

$$S_{bc} = \frac{1}{N_b} \sum_{i=1}^{\ell} \sum_{j=1}^{\ell} K_{(i,j)} I_{\{C_{(i,j)} < 0\}}, \quad (8)$$

where N_w, N_b are the normalizers that respectively count the number of positive and negative elements in C.

By definition, S_{wc} denotes the average similarity between users in the same cluster, and S_{bc} denotes the average similarity between users in different clusters. For high quality clusters, it is quite natural to expect similarity within groups are larger and similarity between groups are smaller. Hence we define the *Normalized Similarity Ratio* as

$$NSR = \frac{S_{wc}}{S_{bc} + \alpha}, \quad (9)$$

where α is a small constant in case $S_{bc} = 0$, which happens frequently on sparse views.

It is noteworthy that, we do not directly penalize imbalance cluster results, since in applications some social circles, such as family, are indeed smaller than other circles, such as friends. However, our metric will lower the score when a super-large cluster appears.

To evaluate the performance over all views, we define the total similarity ratio. Let $S_{wc[j]}$ and $S_{bc[j]}$ respectively denote the within-cluster and between-cluster similarity ratio of view j, where $j \in 1, 2, ..., \ell$, we define the total similarity ratio on one data set as

$$NSR_T = \frac{\sum_{j=1}^{\ell} S_{wc[j]}}{\sum_{j=1}^{\ell} S_{bc[j]}} \quad (10)$$

4.2 Data Collection and View Construction

At present, Facebook and Twitter are two most popular social networking sites, judging by number of active users and daily traffic. Since Facebook users mostly use real identities, it enforces constraints that hinder us from collecting large amount of data. In this research, we collected our data set from Twitter, which is most recognizable for the "tweet" function – the microblog service.

We have implemented a crawler to collect Twitter data using its API, which allows us to get a Twitter user's pro-

file, follower/following lists, and tweet messages. We start with a random user as the seed, and crawl all his/her information (profile, follower/following lists and most recent 2,000 tweets). The intersection of the *follower list* and the *following list* are regarded as *friends*. We crawl the same set of information from the seed user's friends. All the collected data about a seed user and all his/her friends is considered as one *data set*. For each user, we attempt to collect the following information: user name, screen name, user id, profile create time, description (a personal statement), list of followers, list of followings, location and time zone. Meanwhile, for each tweet, we collect the following: tweet id, post time, tweet location, in-reply-to user id, in-reply-to status id, list of re-tweets (user id and tweet id) and tweet content. Note that not all the attributes are available and accurate for all the users. For example, user location in user profiles is self-generated textual description, where we have seen "Worldwide", and "Coming Soon Everywhere" etc. Meanwhile, tweet locations are accurate latitudes and longitudes, but they are missing from most of the tweets.

Twitter has enforced mandatory limits on the crawling rate, especially for crawling account-specific information. We have collected 92 data sets – 92 seed users and all their friends. In our data set, each seed user has 245 friends on average. In total, we have collected information of more than 22K users, with approximately 3 million friendship links, and more than 27 million tweet messages.

We construct six views, as introduced in Section 2: content (V_1), friendship (V_2), common friends (V_3), reply (V_4), re-tweet (V_5) and co-reply (V_6). For each data set, we have n users in total, and use $V_{k,[i,j]}$ to denote the similarity between node N_i and node N_j in view V_k. In particular, for the content view, we have set a threshold of $\rho = 0.2$ to remove 80% of the low-confidence tags (Pareto principle, a.k.a. 80-20 rule). Meanwhile, all the matricides are normalized by dividing every element by the maximum value in the matrix. Then all diagonal elements are set to one, indicating that self-similarity is always the highest among all.

4.3 Experiment Design

We first implement the Selective Co-trained Spectral Clustering ($SCSC$) algorithm with six views, as described in Section 3. After the update iterations in the algorithm, we concatenate U_j's of the most informative views to obtain matrix V, and run k-means on rows of V to obtain the final clustering result. In particular, we concatenate the spectral matrices U_j of the content and structure views to obtain

$$V = [U_{content}, U_{friend}, U_{commonfriend}]. \qquad (11)$$

We choose these views because they are denser in information. We did not concatenate interaction views since they may be too sparse to provide accurate information for all users. However, their information have already been (selectively) transferred to the content and structure views through the multi-view algorithms, i.e. CSC and SCSC.

Baseline approaches. For comparison, we also employ three baseline approaches on our data: SCAN, SC, CSC.

SCAN: Structural Clustering Algorithm for Networks, proposed in [45], is based purely on friendship information of social networks. We run SCAN on the friendship view and compare the results with SCSC.

Table 1: Size of Each Cluster in the Clustering Result. *std* stands for standard deviation of all group sizes.

Cluster	1	2	3	4	5	std
SC	8	12	13	25	44	14.6
CSC	16	17	20	24	25	4.04
$SCSC$	10	10	13	15	54	18.9

Spectral Clustering: SC utilizes eigenvectors and eigenvalues of similarity matrices (or derived matrices), to find the membership for each vertex. We run spectral clustering on each view separately to obtain eigenvectors U_j's. We then column-wise concatenate U_j's of the most informative views to obtain matrix V, and run k-means on rows of V.

CSC (Co-trained Spectral Clustering:) Special type of spectral clustering that exploits multiple sources of information, as mentioned in Section 3. We first run CSC on six views. After the update iterations are done, we concatenate U_j's of the most informative views to obtain matrix V. Run k-means on rows of V.

In the experiments, we observed similar trends of all approaches when changing the number of clusters k from 3 to 10. Hence we set $k = 5$ for all approaches except SCAN. We use default parameters for SCAN.

4.4 Results and Performance Analysis

We first examine the performance of SC, CSC and SCSC approaches on six views of one data set, which contains 386 users. We iterate CSC and $SCSC$ for 20 times and report their normalized similarity ratio (NSR) on each view in Figure 2. We see a general trend that CSC improves its performance as more iterations are done. This coincides with the spirit of co-trained style algorithms. However, the convergence rate is relatively slow and improvements are not very significant on Tag and Reply views. On the Friend and Co-reply views, CSC does not improve the performance of single-view clustering. As we explained before, this may due to the ignorance of CSC on inconsistency between views, especially sparse views. On the other hand, our $SCSC$ approach efficiently and significantly boosts the performance after just one or two iterations. On Common Friend view, we observe a degeneration of $SCSC$, which may be because this view has lower correlation to other views.

We further examine the *balance* of the output clusters produced by each algorithm at their best iterations. An iteration is called the *best iteration of an algorithm* if the algorithm reaches the highest total similarity ratio across all iterations. In Table 1 we summarize the size of each group generated by one algorithm. It can be seen that CSC encourages more balanced clusters, and both SC and $SCSC$ outputs one big clusters. From the algorithmic point of view, this may because CSC enforces stronger consistency across views; hence making the similarity matrix of each view smoother than before. In practice, we think imbalance clusters are acceptable in many applications. For example, a family circle is usually much smaller than a friend circle.

Next we evaluate the performance of all approaches on 92 data sets. The total similarity ratio of each data set is shown in Figure 3 (data sets ordered by the total similarity ratio (TSR) from SCSC). It is clear that SCSC outperforms single-view spectral clustering (SC), while CSC performs the worst. This coincides with our observation in Figure 2, as

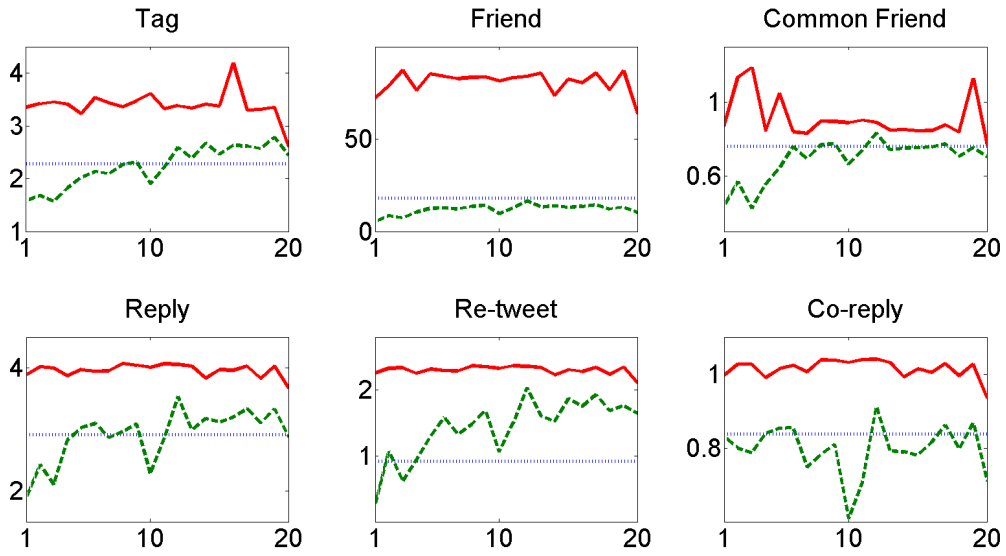


Figure 2: Normalized Similarity Ratio on Six Views. In each figure, the y-axis represents NSR and x-axis represents the number of update iterations. Blue dot curve represents SC approach, green dash curve represents CSC and red solid curve represents our $SCSC$ approach.

well as our analysis of the limitations of CSC: enforcing the complete similarity information to transfer from one view to another may contaminate other views and worsen the performance. Finally, the Mean TSR (MTSR) for SC is 108.8, MTSR for CSC is 24.8, and MTSR for SCSC is 187.4.

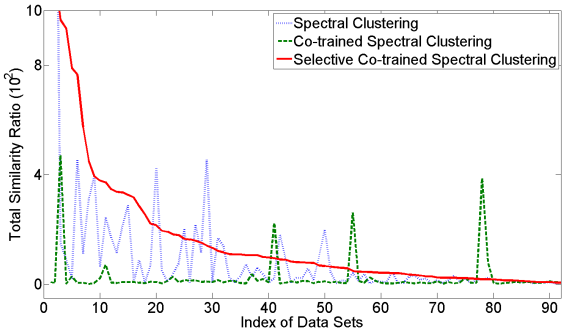


Figure 3: Total similarity ratio (TSR) of all data sets.

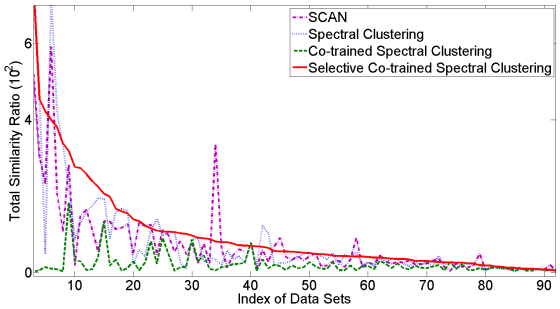


Figure 4: Normalized Similarity Ratio of all Seed Users on Friend View.

Last, we compare all approaches with SCAN, which is designed for structure-based clustering. Since SCAN filters out *outliers*, we evaluate all approaches only on the non-outlier users, to be fair. Total similarity ratio of all data sets are shown in Figure 4. In particular, the ATSR for SCAN is 71.9, while the updated ATSR for SC is 79.5, ATSR for CSC is 20.9, and ATSR for SCSC is 100.6. It is clear that the performance of $SCAN$ is worse than either SC or $SCSC$, but better than CSC.

4.5 Manual Evaluation

Ultimately, the quality of the discovered social circles must be assessed by users. To include users in the loop, we launch a manual evaluation for boundary nodes. As it is impractical to manually examine all users, we attempt to evaluate the nodes that are most doubtful in the clustering process. A boundary node N represents a user who is clustered by SCSC into cluster C_i, but is far away from the centroid of the cluster. In particular, we select the boundary node with the largest distance (i.e., least similarity) from each data set. For each selected N, we identify the cluster C_j, which is (on average) the closest to N other than C_i. We ask users to evaluate if N should be clustered into C_i or C_j.

In the evaluation, we randomly select 5 nodes from cluster i and j, respectively. For each selected node n_k, we display it with N to an external evaluator, and ask the evaluator to answer the question "Do you think N should be in the same social circle as n_k?" In particular, each evaluator marks the node pair (N, n_k) with a score from 1 to 5: 5: strongly agree – they belong to the same circle; 4: somewhat agree; 3: neutral; 2: somewhat disagree; and 1: strongly disagree – they do not belong to the same circle. Please note that the evaluation is *blind*. That is, the evaluators do not know whether the pair of nodes are clustered into the same circle or not. In the experiment, we asked 5 external evaluators (not the authors) to evaluate 60 boundary nodes, which means examining 600 node pairs. As a result, node pairs from the same cluster, as identified by SCSC, earned an average score of 2.63, while node pairs from different circles earned an average score of 2.52.

Table 2: Representative tags for clusters of a seed

Cluster	Representative Tags
C_1	Human,Sleep
C_2	Valentine's Day,Dance,Sport
C_3	Ireland,Beer,Coffee
C_4	Social media,Health,Cancer
C_5	Yahoo!,WHATS'On (Software),Android

From the experiment results, we can conclude that our multi-view clustering approach is effective in clustering users' ego networks into circles. Although the margin appears to be very small, however, we would like to emphasize that we have selected the boundary nodes (N) that SCSC is least confident with in the evaluation. Therefore, the result appears to be acceptable.

4.6 Keyword Extraction for Clusters

To have a direct perception on the content of the circles, we attempt to discover the most unique tags for each cluster. To do so, we calculate the probability of "representativeness" for each tag in each cluster. Intuitively, a tag with larger bias towards a cluster better represents the content of the cluster. Formally, the probability of tag t in cluster C, denoted by $P(t|C)$ can be defined as:

$$P(t|C) = \frac{\sum_{i \in C} tf_{norm}(i,t)}{|C|}$$

$$tf_{norm}(i,t) = \frac{tf(i,t)}{max\{tf(i,t)|t \in \mathbf{T_i}\}}$$

$tf(i,t)$ is the frequency of tag t in user i's content, and the max function returns the largest frequency for all tags in i's content. To find the most representative tags in a certain cluster, we propose to utilize *Kullback-Leibler Divergence* (KL-divergence). In particular, we first construct 2 discrete probability distributions $P_t(i)$ and $Q_t(i)$ as.

$$P_t(i) = \frac{P(t|C_i)}{\sum_{C_i \in \mathbb{C}} P(t|C_i)}$$

$$Q_t(i) = \frac{1}{|\mathbb{C}|}$$

We further calculate the bias of tags:

$$Bias_{KL}(t) = \sum_i (P_t(i) \ln \frac{P_t(i)}{Q_t(i)})$$

For each cluster C_i, we can find the tags with largest $Bias_{KL}$, and having max $P(t|C)$ in C_i as the representative tags for C_i. The top 3 tags for 5 clusters of a randomly selected data set are shown in Table 2. For clusters having less than 3 representative tags, we just show all of them. From this example, we can see different groups have different topics. For instance, group 2 is leaning towards entertainment, group 4 seems to be interested in health care information, while group 5 is quite technical. The extracted content has been confirmed by our manual examination of the circles. As a result, we can actually perceive the separations of different circles in the ego network.

5. RELATED WORK

Identifying social circles from a user's online social networks is important for the individual to exert appropriate access control on information sharing[39] . However, manually managing groups on social network sites might present a burden for users[23, 18], which triggered the idea on using automatic sociocentric network clustering algorithms [14, 18]. Sociocentric network clustering, which is usually referred to as community detection, aims to divide people into groups within which they are more similar[1] and have more connections[32] or relationships. Traditional personal network studies mostly focus on attribute-based data such as age, sex [30]. Meanwhile, most of graph-based methods in community detection (survey [13]) only consider topological structure and linkage information, e.g., graph partitioning [19] and hierarchical clustering [15, 32], maximization of a likelihood [31, 17], matrix factorization [33, 48], etc. There is a trend in recent research based on graphs which combined link information and content or attribute information [46, 26, 35] or interaction information between individuals [51]. Another class of approaches attach greater importance to content or link context information. [9, 27, 49, 50] use methods like topic modeling to take full advantage of semantic information, such as email, tweet messages, and documents, in detecting communities from a social network. [44] proposed a method to find like-minded people who share more semantically relevant tags. A recent research [36] propose generative Bayesian models to utilize not only topics and social graph topology but also nature of user interactions to discover latent communities in social graphs. The difference between their work and ours is that we formalize different types of views, and also use content annotation on the content view, which concerns the understanding of the information and is more meaningful in finding similar topics. Meanwhile, our clustering algorithm is completely different from [36].

With the rapid growth of online social networks, privacy concerns of personal information arise, e.g., [16, 28, 47]. Based on privacy concerns, [29] developed a model to discover social circles by using both network structure and user profile information; [40] proposed an approach based on apriori algorithm to identify hidden groups by dynamically detecting grouping criteria, i.e. certain combinations of properties of a user's contacts, such as relationship, location, hobbies, age, privacy, etc. The difficulty in utilizing this kind of methods is that automatically collecting attributes of users through online social network is a nontrivial task although traditional personal network studies can collect these information through interviews more easily.

Algorithmically, we employ the multi-view clustering framework to detect social circles considering the multi-view nature of an ego network. This framework provides an automatic way of merging information from multiple sources, and has demonstrated superior performance in many applications such as document categorization [5], digit classification [21, 22] or image annotation [43]. In this paper we base our analysis on co-trained spectral clustering [21], for it uses the similarity matrices as input, which coincides with our application setting.

Co-trained spectral clustering (CSC) is an algorithm that clusters data with multiple views. It is an extension of the well-known spectral clustering algorithm [37], which groups data in the spectral embedded space. It has been shown that the embedded space contains discriminative information for the underlying structure of a data set. CSC inherits the iterative nature of co-training [6], i.e., it alternately projects

data in one view into the combined spectral spaces of other views and then groups them using standard k-means clustering algorithm. Such cross-view projection allows the discriminative information of other views to be implicitly transferred to the current view. While this framework is computationally efficient, its implicit schema hinders an explicit control of the information transferred across views. Moreover, it treats all views equally whereas in many applications the exposition or quality of different views are different and, as a consequence, negative transfer may occur. For example, in our ego network the interaction views are usually too sparse to be completely consistent with other views due to its low frequency of usage, which means it they should not be treated equally as all other views. These limitations of CSC are tackled in our proposed framework, which uses the combined clustering result of other views to refine the current view. This schema allows an explicit and selective transfer of information across views based on each pair of users, and leads to faster convergence of learning, as demonstrated in our empirical study.

6. DISCUSSIONS

Computational complexity of SCSC. For an ego network with n users and ℓ views, suppose all users are clustered into k groups by an iterative multi-view clustering algorithm, i.e. CSC or SCSC, which updates for t rounds and finally applies standard k-means. The computational complexity of CSC is $O(\ell t + nk^2)$ and that of SCSC is $O(\ell tnk^2 + nk^2)$, where the extra overload $O(nk^2)$ of SCSC arises from the k-means clustering performed in each round of update [2]. At a first glance, SCSC suffers a much heavier computational burden than CSC. However, we argue that in practice such additional cost is quite tolerable.

First of all, we emphasize the difference between an ego network and a general social network: the former consists of very limited users (a typical ego network we crawled has around 200 friends) and grows slowly, whereas the latter usually consists of millions of users and scales up quickly. This implies both n and k will remain small and not bring in too much extra computation. Second, the updates of each view in one round are independent and thus can be parallelized. This prevents the computational burden from being accumulated over views, and reduces the complexity of SCSC to $O(tnk^2 + nk^2)$. Besides, as shown in our experiments, SCSC converges much faster than CSC and thus needs much smaller t rounds of update in practice. By trading some computational efficiency for adaptiveness, SCSC significantly boosts the clustering performance.

Non-overlapping vs. overlapping circles. In the ego network E_S of seed S, if we allow any user N_i to belong to multiple circles, it is regarded as *overlapping circles*. Meanwhile, if each user N_i is allowed in exactly one circle, it is *non-overlapping circles*. In the literature, both types of circles have been used. In this paper, we select non-overlapping circles for two major reasons. First, our approach is primarily motivated by privacy protection and information boundary enforcement in social networks. When two social circles in the ego network overlaps, the overlapping users observe information from both circles. Such users may also easily violate the boundaries by moving information from its origin circle to the other overlapping circles. This is the online version of "social gossip". On the other hand, in theory, overlapping and non-overlapping circles are essentially equivalent. That is, two overlapping circles A and B could be converted to three non-overlapping circles $A \cap B; A \setminus B; B \setminus A$. Contained circles $A \subset B$ could be converted to two non-overlapping circles $A; B \setminus A$.

Applications of social circles. As suggested in [40, 41], social circles are used to protect information privacy, by delivering messages to designated circles and enforcing circle boundaries. Automatically clustered circles are presented to users, so that they could further re-organize and configure such circles. In socialization, messages are posted to the selected circles. Meanwhile, social circle enforcement becomes particularly challenging when some social networking sites allows breaches in privacy protection (e.g., when users are allowed to "re-share" private posts of their friends). However, those issues are outside of the scope of this paper.

On the other hand, the discovered social circles could be used to improve the efficiency of ad delivery, targeted advertising, and opinion mining in social groups. Social circles could also be used to study users' socialization behavior and social network information flow. If temporal information is added to the data, we can extend our model to further study the development of social circles and evolution of ego networks.

7. CONCLUSION

With the extreme popularity of online social networks, privacy becomes a major concern. The notions of social circles and information boundary have been proposed, to protect private information and to facilitate secure socialization. However, the problem of social circle discovery remains open and challenging. In this paper, we start with our observations that users belonging to the same circle are very likely to: (1) be friends and share many common friends; (2) be interest in similar content; (3) have more interactions with each other. We model the ego network with 6 different views, and we argue that features from different views would complement each other. We propose an automatic social circle detection mechanism utilizing multi-view clustering. In particular, we propose a one-side co-trained spectral clustering technique, which is tailored for the sparse nature of our data. We tested our algorithms with real-world social networking data collected from Twitter. Experiment results show that our approach is both effective and efficient.

Acknowledgments

The work has been in part supported by NSF CNS-1422206, NSF CNS-1337899, NSF OIA-1028098, and the University of Kansas General Research Fund (GRF).

8. REFERENCES

[1] L. A. Adamic and E. Adar. Friends and neighbors on the web. *Social Networks*, 25:211–230, 2003.
[2] K. Alsabti, S. Ranka, and V. Singh. An efficient k-means clustering algorithm. 1997.
[3] S. B. Barnes. A privacy paradox: Social networking in the united states. *First Monday [Online]*, 11(9), 2006.
[4] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE T-PAMI*, 19(7), 1997.

[2]For more information about the efficiency of k-means, readers are referred to [2].

[5] S. Bickel and T. Scheffer. Multi-view clustering. In *ICDM*, volume 4, pages 19–26, 2004.

[6] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100. ACM, 1998.

[7] A. B. Brush, J. Krumm, and J. Scott. Exploring end user preferences for location obfuscation, location-based services, and the value of location. In *Ubicomp*, 2010.

[8] H.-W. Chang, D. Lee, M. Eltaher, and J. Lee. @phillies tweeting from philly? predicting twitter user locations with spatial word usage. In *ASONAM*, 2012.

[9] J. Chang, J. Boyd-Graber, and D. M. Blei. Connections between the lines: augmenting social networks with text. *KDD'09*, pages 169–178, 2009.

[10] Z. Cheng, J. Caverlee, and K. Lee. You are where you tweet: a content-based approach to geo-locating twitter users. pages 759–768, 2010.

[11] M. Culnan and J. Bies. Consumer privacy: Balancing economic and justice considerations. *Journal of Social Issues*, 59(2), 2003.

[12] P. Ferragina and U. Scaiella. Fast and accurate annotation of short texts with wikipedia pages. *Software, IEEE*, 29(1):70–75, 2012.

[13] S. Fortunato. Community detection in graphs. *Physics Reports*, 486:75–174, 2010.

[14] E. Gilbert and K. Karahalios. Predicting tie strength with social media. *CHI'09*, pages 211–220, 2009.

[15] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proc. of the National Academy of Sciences*, 99:7821–7826, 2002.

[16] R. Gross and A. Acquisti. Information revelation and privacy in online social networks. In *Proceedings of the 2005 ACM workshop on Privacy in the electronic society*, pages 71–80. ACM, 2005.

[17] J. M. Hoffman and C. H. Wiggins. A bayesian approach to network modularity. *Phys. Rev. Lett.*, 100:258701, 2008.

[18] S. Jones and E. O'Neill. Feasibility of structural network clustering for group-based privacy control in social networks. *SUPS*, July 2010.

[19] B. W. Kernighan and S. Lin. An efficient heuristic procedure for partitioing graphs. *The Bell System Tehnical Journal*, 49:291–307, 1970.

[20] J. Krumm. A survey of computational location privacy. *Personal Ubiquitous Comput.*, 13(6), 2009.

[21] A. Kumar and H. D. Iii. A co-training approach for multi-view spectral clustering. In *ICML-11*, 2011.

[22] A. Kumar, P. Rai, and H. D. Iii. Co-regularized multi-view spectral clustering. In *NIPS*, pages 1413–1421, 2011.

[23] S. Lederer, J. I. Hong, A. K. Dey, and J. A. Landay. Personal privacy through understanding and action: five pitfalls for designers. *Personal and Ubiquitous Computing*, 8(6):440–454, 2004.

[24] K. Lewis, J. Kaufman, and N. Christakis. The taste for privacy: An analysis of college student privacy settings in an online social network. *Journal of Computer-Mediated Communication*, 14(1), 2008.

[25] W. Li, P. Serdyukov, A. P. de Vries, C. Eickhoff, and M. Larson. The where in the tweet. In *CIKM*, 2011.

[26] W. Lin, X. Kong, P. S. Yu, Q. Wu, Y. Jia, and C. Li. Community detection in incomplete information networks. *WWW'12*, April 2012.

[27] Y. Liu, A. N. Mizil, and W. Gryc. Topic-link lda: Joint models of topic and author community. *ICML'09*, 382:665–672, 2009.

[28] B. Luo and D. Lee. On protecting private information in social networks: a proposal. In *IEEE ICDE Workshop on Modeling, Managing, and Mining of Evolving Social Networks (M3SN)*. IEEE, 2009.

[29] J. McAuley and J. Leskovec. Discovering social circles in ego networks. *TKDD'13*, 2012.

[30] C. McCarty. Structure in personal networks. *Journal of Social Structure*, 3, 2002.

[31] M. E. J. Newman and E. A. Leicht. Mixture models and exploratory analysis in networks. *Proc Natl Acad Sci USA*, 104:9564ÍC9569, June 2007.

[32] M. E. J. Newman and M.Girvan. Finding and evaluating community structure in networks. *Rhys. Reve. E*, 69:026113, Feb 2004.

[33] I. Psorakis, S. Roberts, M. Ebden, and B. Sheldon. Overlapping community detection using bayesian non-negative matrix factorization. *Physical Review E*, 83:066114, June 2011.

[34] H. Qian and C. R. Scott. Anonymity and self-disclosure on weblogs. *Journal of Computer-Mediated Communication*, 12(4), 2007.

[35] Y. Ruan, D. Fuhry, and S. Parthasarathy. Efficient community detection in large networks using content and links. *WWW'13*, pages 1089–1098, May 2013.

[36] M. Sachan, D. Contractor, T. A. Faruquie, and L. V. Subramaniam. Using content and interactions for discovering communities in social networks. *WWW'12*, pages 331–340, April 2012.

[37] J. Shi and J. Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8), 2000.

[38] P. Shi, H. Xu, and Y. Chen. Using contextual integrity to examine interpersonal information boundary on social network sites. In *CHI*.

[39] M. M. Skeels and J. Grudin. When social networks cross boundaries:a case study of workplace use of facebook and linkedin. *GROUP'09*, May 2009.

[40] A. Squicciarini, S. Karumanchi, D. Lin, and N. DeSisto. Identifying hidden social circles for advanced privacy configuration. *Computers & Security*, 2013.

[41] A. Squicciarini, D. Lin, S. Karumanchi, and N. DeSisto. Automatic social group organization and privacy management. In *CollaborateCom*, 2012.

[42] H. Tavani. Philosophical theories of privacy: Implications for an adequate online privacy policy. *Metaphilosophy*, 38(1), 2007.

[43] H. Wang, F. Nie, and H. Huang. Multi-view clustering and feature learning via structured sparsity. 2013.

[44] X. Wang, H. Liu, and W. Fan. Connectiong users with similar interests via tag network inference. *CIKM'11*, pages 1019–1024, October 2011.

[45] X. Xu, N. Yuruk, Z. Feng, and T. A. Schweiger. Scan: a structural clustering algorithm for networks. In *ACM SIGKDD*. ACM, 2007.

[46] T. Yang, R. Jin, Y. Chi, and S. Zhu. Combining link and content for community detection:a discriminative approach. *KDD'09*, pages 927–936, 2009.

[47] Y. Yang, J. Lutes, F. Li, B. Luo, and P. Liu. Stalking online: on user privacy in social networks. In *ACM conference on Data and Application Security and Privacy (CODASPY)*, 2012.

[48] Y. Zhang and D.-Y. Yeung. Overlapping community detection via bounded nonnegative matrix tri-factorization. *KDD*, August 2012.

[49] D. Zhou, E. Manavoglu, J. Li, C. L. Giles, and H. Zha. Probabilistic models for discovering e-communities. *WWW'06*, pages 173–182, 2006.

[50] W. Zhou, H. Jin, and Y. Liu. Community discovery and profiling with social messages. *KDD'12*, pages 388–396, August 2012.

[51] Y. Zhou and L. Liu. Social influence based clustering of heterogeneous. *KDD'13*, August 2013.