Learning to Discover Social Circles in Ego Networks

Julian McAuley
Stanford, USA
jmcauley@cs.stanford.edu

Jure Leskovec
Stanford, USA
jure@cs.stanford.edu

Abstract

Our personal social networks are big and cluttered, and currently there is no good way to organize them. Social networking sites allow users to manually categorize their friends into *social circles* (e.g. 'circles' on Google+, and 'lists' on Facebook and Twitter), however they are laborious to construct and must be updated whenever a user's network grows. We define a novel machine learning task of identifying users' social circles. We pose the problem as a node clustering problem on a user's ego-network, a network of connections between her friends. We develop a model for detecting circles that combines network structure as well as user profile information. For each circle we learn its members and the circle-specific user profile similarity metric. Modeling node membership to multiple circles allows us to detect overlapping as well as hierarchically nested circles. Experiments show that our model accurately identifies circles on a diverse set of data from Facebook, Google+, and Twitter for all of which we obtain hand-labeled ground-truth.

1 Introduction

Online social networks allow users to follow streams of posts generated by hundreds of their friends and acquaintances. Users' friends generate overwhelming volumes of information and to cope with the 'information overload' they need to organize their personal social networks. One of the main mechanisms for users of social networking sites to organize their networks and the content generated by them is to categorize their friends into what we refer to as *social circles*. Practically all major social networks provide such functionality, for example, 'circles' on Google+, and 'lists' on Facebook and Twitter. Once a user creates her circles, they can be used for content filtering (e.g. to filter status updates posted by distant acquaintances), for privacy (e.g. to hide personal information from coworkers), and for sharing groups of users that others may wish to follow.

Currently, users in Facebook, Google+ and Twitter identify their circles either manually, or in a naïve fashion by identifying friends sharing a common attribute. Neither approach is particularly satisfactory: the former is time consuming and does not update automatically as a user adds more friends, while the latter fails to capture individual aspects of users' communities, and may function poorly when profile information is missing or withheld.

In this paper we study the problem of automatically discovering users' social circles. In particular, given a single user with her personal social network, our goal is to identify her circles, each of which is a subset of her friends. Circles are user-specific as each user organizes her personal network of friends independently of all other users to whom she is not connected. This means that we can formulate the problem of circle detection as a clustering problem on her ego-network, the network of friendships between her friends. In Figure 1 we are given a single user u and we form a network between her friends v_i. We refer to the user u as the *ego* and to the nodes v_i as *alters*. The task then is to identify the circles to which each alter v_i belongs, as in Figure 1. In other words, the goal is to find nested as well as overlapping communities/clusters in u's ego-network.

Generally, there are two useful sources of data that help with this task. The first is the set of edges of the ego-network. We expect that circles are formed by densely-connected sets of alters [20].

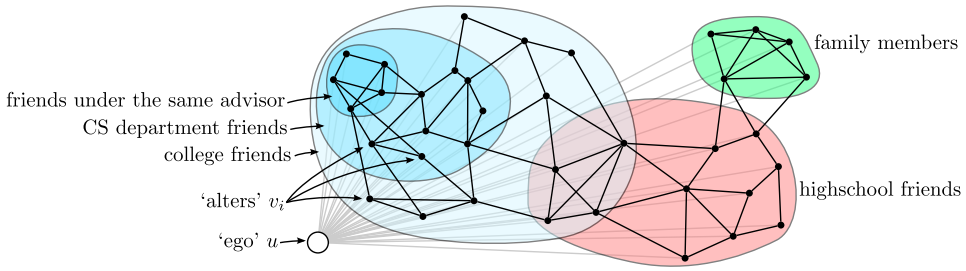

Figure 1: An ego-network with labeled circles. This network shows typical behavior that we observe in our data: Approximately 25% of our ground-truth circles (from Facebook) are contained *completely* within another circle, 50% overlap with another circle, and 25% of the circles have no members in common with any other circle. The goal is to discover these circles given only the network between the ego's friends. We aim to discover circle memberships and to find common properties around which circles form.

However, different circles overlap heavily, i.e., alters belong to multiple circles simultaneously [1, 21, 28, 29], and many circles are hierarchically nested in larger ones (Figure 1). Thus it is important to model an alter's memberships to multiple circles. Secondly, we expect that each circle is not only densely connected but its members also share common properties or traits [18, 28]. Thus we need to explicitly model different dimensions of user profiles along which each circle emerges.

We model circle affiliations as latent variables, and similarity between alters as a function of common profile information. We propose an *unsupervised* method to learn which dimensions of profile similarity lead to densely linked circles. Our model has two innovations: First, in contrast to mixed-membership models [2] we predict *hard* assignment of a node to *multiple* circles, which proves critical for good performance. Second, by proposing a parameterized definition of profile similarity, we learn the dimensions of similarity along which links emerge. This extends the notion of homophily [12] by allowing different circles to form along different social dimensions, an idea related to the concept of Blau spaces [16]. We achieve this by allowing each circle to have a different definition of profile similarity, so that one circle might form around friends from the same school, and another around friends from the same location. We learn the model by simultaneously choosing node circle memberships and profile similarity functions so as to best explain the observed data.

We introduce a dataset of 1,143 ego-networks from Facebook, Google+, and Twitter, for which we obtain hand-labeled ground-truth from 5,636 different circles.[1] Experimental results show that by simultaneously considering social network structure as well as user profile information our method performs significantly better than natural alternatives and the current state-of-the-art. Besides being more accurate our method also allows us to generate automatic explanations of why certain nodes belong to common communities. Our method is completely unsupervised, and is able to automatically determine both the number of circles as well as the circles themselves.

Further Related Work. Topic-modeling techniques have been used to uncover 'mixed-memberships' of nodes to multiple groups [2], and extensions allow entities to be attributed with text information [3, 5, 11, 13, 26]. Classical algorithms tend to identify communities based on node features [9] or graph structure [1, 21], but rarely use both in concert. Our work is related to [30] in the sense that it performs clustering on social-network data, and [23], which models memberships to multiple communities. Finally, there are works that model network data similar to ours [6, 17], though the underlying models do not form communities. As we shall see, our problem has unique characteristics that require a new model. An extended version of our article appears in [15].

2 A Generative Model for Friendships in Social Circles

We desire a model of circle formation with the following properties: (1) Nodes within circles should have common properties, or 'aspects'. (2) Different circles should be formed by different aspects, e.g. one circle might be formed by family members, and another by students who attended the same university. (3) Circles should be allowed to overlap, and 'stronger' circles should be allowed to form within 'weaker' ones, e.g. a circle of friends from the same degree program may form within a circle

[1] `http://snap.stanford.edu/data/`

from the same university, as in Figure 1. (4) We would like to leverage both profile information and network structure in order to identify the circles. Ideally we would like to be able to pinpoint *which* aspects of a profile caused a circle to form, so that the model is interpretable by the user.

The input to our model is an ego-network $G = (V, E)$, along with 'profiles' for each user $v \in V$. The 'center' node u of the ego-network (the 'ego') is not included in G, but rather G consists only of u's friends (the 'alters'). We define the ego-network in this way precisely because creators of circles do not themselves appear in their own circles. For each ego-network, our goal is to predict a set of circles $\mathcal{C} = \{C_1 \dots C_K\}$, $C_k \subseteq V$, and associated parameter vectors θ_k that encode how each circle emerged. We encode 'user profiles' into pairwise features $\phi(x, y)$ that in some way capture what properties the users x and y have in common. We first describe our model, which can be applied using arbitrary feature vectors $\phi(x, y)$, and in Section 5 we describe several ways to construct feature vectors $\phi(x, y)$ that are suited to our particular application.

We describe a model of social circles that treats circle memberships as latent variables. Nodes within a common circle are given an opportunity to form an edge, which naturally leads to hierarchical and overlapping circles. We will then devise an *unsupervised* algorithm to jointly optimize the latent variables and the profile similarity parameters so as to best explain the observed network data.

Our model of social circles is defined as follows. Given an ego-network G and a set of K circles $\mathcal{C} = \{C_1 \dots C_K\}$, we model the probability that a pair of nodes $(x, y) \in V \times V$ form an edge as

$$p((x,y) \in E) \propto \exp\left\{ \underbrace{\sum_{C_k \supseteq \{x,y\}} \langle \phi(x,y), \theta_k \rangle}_{\text{circles containing both nodes}} - \underbrace{\sum_{C_k \not\supseteq \{x,y\}} \alpha_k \langle \phi(x,y), \theta_k \rangle}_{\text{all other circles}} \right\}. \tag{1}$$

For each circle C_k, θ_k is the profile similarity parameter that we will learn. The idea is that $\langle \phi(x, y), \theta_k \rangle$ is high if both nodes belong to C_k, and low if either of them do not (α_k trades-off these two effects). Since the feature vector $\phi(x, y)$ encodes the similarity between the profiles of two users x and y, the parameter vector θ_k encodes what dimensions of profile similarity caused the circle to form, so that nodes within a circle C_k should 'look similar' according to θ_k.

Considering that edges $e = (x, y)$ are generated independently, we can write the probability of G as

$$P_\Theta(G; \mathcal{C}) = \prod_{e \in E} p(e \in E) \times \prod_{e \notin E} p(e \notin E), \tag{2}$$

where $\Theta = \{(\theta_k, \alpha_k)\}^{k=1\dots K}$ is our set of model parameters. Defining the shorthand notation

$$d_k(e) = \delta(e \in C_k) - \alpha_k \delta(e \notin C_k), \quad \Phi(e) = \sum_{C_k \in \mathcal{C}} d_k(e) \langle \phi(e), \theta_k \rangle$$

allows us to write the log-likelihood of G:

$$l_\Theta(G; \mathcal{C}) = \sum_{e \in E} \Phi(e) - \sum_{e \in V \times V} \log\left(1 + e^{\Phi(e)}\right). \tag{3}$$

Next, we describe how to optimize node circle memberships $\mathcal{C}$ as well as the parameters of the user profile similarity functions $\Theta = \{(\theta_k, \alpha_k)\}$ ($k = 1 \dots K$) given a graph G and user profiles.

3 Unsupervised Learning of Model Parameters

Treating circles $\mathcal{C}$ as latent variables, we aim to find $\hat{\Theta} = \{\hat{\theta}, \hat{\alpha}\}$ so as to maximize the regularized log-likelihood of (eq. 3), i.e.,

$$\hat{\Theta}, \hat{\mathcal{C}} = \underset{\Theta, \mathcal{C}}{\operatorname{argmax}}\, l_\Theta(G; \mathcal{C}) - \lambda \Omega(\theta). \tag{4}$$

We solve this problem using coordinate ascent on Θ and $\mathcal{C}$ [14]:

$$\mathcal{C}^t = \underset{\mathcal{C}}{\operatorname{argmax}}\, l_{\Theta^t}(G; \mathcal{C}) \tag{5}$$

$$\Theta^{t+1} = \underset{\Theta}{\operatorname{argmax}}\, l_\Theta(G; \mathcal{C}^t) - \lambda \Omega(\theta). \tag{6}$$

Noting that (eq. 3) is concave in θ, we optimize (eq. 6) through gradient ascent, where partial derivatives are given by

$$\begin{aligned}\frac{\partial l}{\partial \theta_k} &= \sum_{e \in V \times V} -d_e(k)\theta_k \frac{e^{\Phi(e)}}{1+e^{\Phi(e)}} + \sum_{e \in E} d_k(e)\theta_k - \frac{\partial \Omega}{\partial \theta_k} \\ \frac{\partial l}{\partial \alpha_k} &= \sum_{e \in V \times V} \delta(e \notin C_k) \left\langle \phi(e), \theta_k \right\rangle \frac{e^{\Phi(e)}}{1+e^{\Phi(e)}} - \sum_{e \in E} \delta(e \notin C_k) \left\langle \phi(e), \theta_k \right\rangle .\end{aligned}$$

For fixed $\mathcal{C} \setminus C_i$ we note that solving $\operatorname{argmax}_{C_i} l_\Theta(G; \mathcal{C} \setminus C_i)$ can be expressed as pseudo-boolean optimization in a pairwise graphical model [4], i.e., it can be written as

$$C_k = \operatorname*{argmax}_C \sum_{(x,y) \in V \times V} E_{(x,y)}(\delta(x \in C), \delta(y \in C)). \tag{7}$$

In words, we want edges with high weight (under θ_k) to appear in C_k, and edges with low weight to appear outside of C_k. Defining $o_k(e) = \sum_{C_k \in \mathcal{C} \setminus C_i} d_k(e) \left\langle \phi(e), \theta_k \right\rangle$ the energy E_e of (eq. 7) is

$$\begin{aligned}E_e(0,0) = E_e(0,1) = E_e(1,0) &= \begin{cases} o_k(e) - \alpha_k \left\langle \phi(e), \theta_k \right\rangle - \log(1 + e^{o_k(e) - \alpha_k \left\langle \phi(e), \theta_k \right\rangle}), & e \in E \\ -\log(1 + e^{o_k(e) - \alpha_k \left\langle \phi(e), \theta_k \right\rangle}), & e \notin E \end{cases} \\ E_e(1,1) &= \begin{cases} o_k(e) + \left\langle \phi(e), \theta_k \right\rangle - \log(1 + e^{o_k(e) + \left\langle \phi(e), \theta_k \right\rangle}), & e \in E \\ -\log(1 + e^{o_k(e) + \left\langle \phi(e), \theta_k \right\rangle}), & e \notin E \end{cases} .\end{aligned}$$

By expressing the problem in this form we can draw upon existing work on pseudo-boolean optimization. We use the publicly-available 'QPBO' software described in [22], which is able to accurately approximate problems of the form shown in (eq. 7). We solve (eq. 7) for each C_k in a random order.

The two optimization steps of (eq. 5) and (eq. 6) are repeated until convergence, i.e., until $\mathcal{C}^{t+1} = \mathcal{C}^t$. We regularize (eq. 4) using the ℓ_1 norm, i.e., $\Omega(\theta) = \sum_{k=1}^{K} \sum_{i=1}^{|\theta_k|} |\theta_{ki}|$, which leads to sparse (and readily interpretable) parameters. Since ego-networks are naturally relatively small, our algorithm can readily handle problems at the scale required. In the case of Facebook, the average ego-network has around 190 nodes [24], while the largest network we encountered has 4,964 nodes. Note that since the method is *unsupervised*, inference is performed independently for each ego-network. This means that our method could be run on the full Facebook graph (for example), as circles are independently detected for each user, and the ego-networks typically contain only hundreds of nodes.

Hyperparameter estimation. To choose the optimal number of circles, we choose K so as to minimize an approximation to the Bayesian Information Criterion (BIC) [2, 8, 25],

$$\hat{K} = \operatorname*{argmin}_K BIC(K; \Theta^K) \tag{8}$$

where Θ^K is the set of parameters predicted for a particular number of communities K, and

$$BIC(K; \Theta^K) \simeq -2 l_{\Theta^K}(G; \mathcal{C}) + |\Theta^K| \log |E|. \tag{9}$$

The regularization parameter $\lambda \in \{0, 1, 10, 100\}$ was determined using leave-one-out cross validation, though in our experience did not significantly impact performance.

4 Dataset Description

Our goal is to evaluate our unsupervised method on ground-truth data. We expended significant time, effort, and resources to obtain high quality hand-labeled data.[2] We were able to obtain ego-networks and ground-truth from three major social networking sites: Facebook, Google+, and Twitter.

From Facebook we obtained profile and network data from 10 ego-networks, consisting of 193 circles and 4,039 users. To do so we developed our own Facebook application and conducted a survey of ten users, who were asked to manually identify all the circles to which their friends belonged. On average, users identified 19 circles in their ego-networks, with an average circle size of 22 friends. Examples of such circles include students of common universities, sports teams, relatives, etc.

[2] `http://snap.stanford.edu/data/`

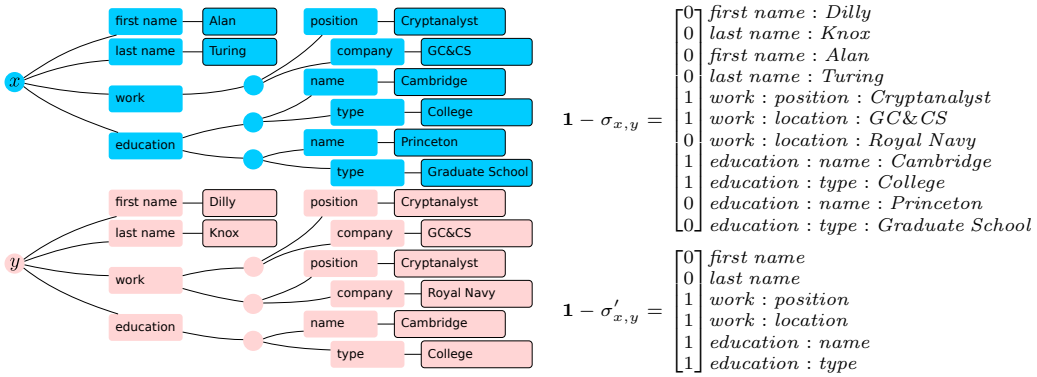

Figure 2: Feature construction. Profiles are tree-structured, and we construct features by comparing paths in those trees. Examples of trees for two users x (blue) and y (pink) are shown at left. Two schemes for constructing feature vectors from these profiles are shown at right: (1) (top right) we construct binary indicators measuring the difference between leaves in the two trees, e.g. 'work→position→Cryptanalyst' appears in both trees. (2) (bottom right) we sum over the leaf nodes in the first scheme, maintaining the fact that the two users worked at the same institution, but discarding the *identity* of that institution.

For the other two datasets we obtained publicly accessible data. From Google+ we obtained data from 133 ego-networks, consisting of 479 circles and 106,674 users. The 133 ego-networks represent all 133 Google+ users who had shared at least two circles, and whose network information was publicly accessible at the time of our crawl. The Google+ circles are quite different to those from Facebook, in the sense that their creators have chosen to release them publicly, and because Google+ is a *directed* network (note that our model can very naturally be applied to both to directed and undirected networks). For example, one circle contains candidates from the 2012 republican primary, who presumably do not follow their followers, nor each other. Finally, from Twitter we obtained data from 1,000 ego-networks, consisting of 4,869 circles (or 'lists' [10, 19, 27, 31]) and 81,362 users. The ego-networks we obtained range in size from 10 to 4,964 nodes.

Taken together our data contains 1,143 different ego-networks, 5,541 circles, and 192,075 users. The size differences between these datasets simply reflects the availability of data from each of the three sources. Our Facebook data is *fully labeled*, in the sense that we obtain *every* circle that a user considers to be a cohesive community, whereas our Google+ and Twitter data is only *partially labeled*, in the sense that we only have access to public circles. We design our evaluation procedure in Section 6 so that partial labels cause no issues.

5 Constructing Features from User Profiles

Profile information in all of our datasets can be represented as a *tree* where each level encodes increasingly specific information (Figure 2, left). From Google+ we collect data from six categories (gender, last name, job titles, institutions, universities, and places lived). From Facebook we collect data from 26 categories, including hometowns, birthdays, colleagues, political affiliations, etc. For Twitter, many choices exist as proxies for user profiles; we simply collect data from two categories, namely the set of hashtags and mentions used by each user during two-weeks' worth of tweets. 'Categories' correspond to parents of leaf nodes in a profile tree, as shown in Figure 2.

We first describe a difference vector to encode the relationship between two profiles. A non-technical description is given in Figure 2. Suppose that users $v \in V$ each have an associated profile tree T_v, and that $l \in T_v$ is a leaf in that tree. We define the difference vector $\sigma_{x,y}$ between two users x and y as a binary indicator encoding the profile aspects where users x and y differ (Figure 2, top right):

$$\sigma_{x,y}[l] = \delta((l \in \mathcal{T}_x) \neq (l \in \mathcal{T}_y)). \tag{10}$$

Note that feature descriptors are defined *per ego-network*: while many thousands of high schools (for example) exist among all Facebook users, only a small number appear among any particular user's friends.

Although the above difference vector has the advantage that it encodes profile information at a fine granularity, it has the disadvantage that it is high-dimensional (up to 4,122 dimensions in the data

we considered). One way to address this is to form difference vectors based on the *parents* of leaf nodes: this way, we encode what profile *categories* two users have in common, but disregard specific values (Figure 2, bottom right). For example, we encode *how many* hashtags two users tweeted in common, but discard *which* hashtags they tweeted:

$$\sigma'_{x,y}[p] = \sum_{l \in children(p)} \sigma_{x,y}[l]. \tag{11}$$

This scheme has the advantage that it requires a *constant* number of dimensions, regardless of the size of the ego-network (26 for Facebook, 6 for Google+, 2 for Twitter, as described above).

Based on the difference vectors $\sigma_{x,y}$ (and $\sigma'_{x,y}$) we now describe how to construct edge features $\phi(x, y)$. The first property we wish to model is that *members of circles should have common relationships* with each other:

$$\phi^1(x, y) = (1; -\sigma_{x,y}). \tag{12}$$

The second property we wish to model is that *members of circles should have common relationships to the ego of the ego-network*. In this case, we consider the profile tree T_u *from the ego user* u. We then define our features in terms of that user:

$$\phi^2(x, y) = (1; -\left|\sigma_{x,u} - \sigma_{y,u}\right|) \tag{13}$$

($|\sigma_{x,u} - \sigma_{y,u}|$ is taken elementwise). These two parameterizations allow us to assess which mechanism better captures users' subjective definition of a circle. In both cases, we include a constant feature ('1'), which controls the probability that edges form within circles, or equivalently it measures the extent to which circles are made up of friends. Importantly, this allows us to predict memberships even for users who have no profile information, simply due to their patterns of connectivity.

Similarly, for the 'compressed' difference vector $\sigma'_{x,y}$, we define

$$\psi^1(x, y) = (1; -\sigma'_{x,y}), \quad \psi^2(x, y) = (1; -\left|\sigma'_{x,u} - \sigma'_{y,u}\right|). \tag{14}$$

To summarize, we have identified four ways of representing the compatibility between different aspects of profiles for two users. We considered two ways of constructing a difference vector ($\sigma_{x,y}$ vs. $\sigma'_{x,y}$) and two ways of capturing the compatibility of a pair of profiles ($\phi(x, y)$ vs. $\psi(x, y)$).

6 Experiments

Although our method is unsupervised, we can evaluate it on ground-truth data by examining the maximum-likelihood assignments of the latent circles $\mathcal{C} = \{C_1 \dots C_K\}$ after convergence. Our goal is that for a properly regularized model, the latent variables will align closely with the human labeled ground-truth circles $\bar{\mathcal{C}} = \{\bar{C}_1 \dots \bar{C}_{\bar{K}}\}$.

Evaluation metrics. To measure the alignment between a predicted circle C and a ground-truth circle $\bar{C}$, we compute the Balanced Error Rate (BER) between the two circles [7], $BER(C, \bar{C}) = \frac{1}{2}\left(\frac{|C \backslash \bar{C}|}{|C|} + \frac{|\bar{C} \backslash C|}{|\bar{C}|}\right)$. This measure assigns equal importance to false positives and false negatives, so that trivial or random predictions incur an error of 0.5 on average. Such a measure is preferable to the $0/1$ loss (for example), which assigns extremely low error to trivial predictions. We also report the F_1 score, which we find produces qualitatively similar results.

Aligning predicted and ground-truth circles. Since we do not know the correspondence between circles in $\mathcal{C}$ and $\bar{\mathcal{C}}$, we compute the optimal match via linear assignment by maximizing:

$$\max_{f: \mathcal{C} \to \bar{\mathcal{C}}} \frac{1}{|f|} \sum_{C \in \text{dom}(f)} (1 - BER(C, f(C))), \tag{15}$$

where f is a (partial) correspondence between $\mathcal{C}$ and $\bar{\mathcal{C}}$. That is, if the number of predicted circles $|\mathcal{C}|$ is less than the number of ground-truth circles $|\bar{\mathcal{C}}|$, then every circle $C \in \mathcal{C}$ must have a match $\bar{C} \in \bar{\mathcal{C}}$, but if $|\mathcal{C}| > |\bar{\mathcal{C}}|$, we do not incur a penalty for additional predictions that *could* have been circles but were not included in the ground-truth. We use established techniques to estimate the number of circles, so that none of the baselines suffers a disadvantage by mispredicting $\hat{K} = |\mathcal{C}|$, nor can any method predict the 'trivial' solution of returning the powerset of all users. We note that removing the bijectivity requirement (i.e., forcing all circles to be aligned by allowing multiple predicted circles to match a single groundtruth circle or *vice versa*) lead to qualitatively similar results.

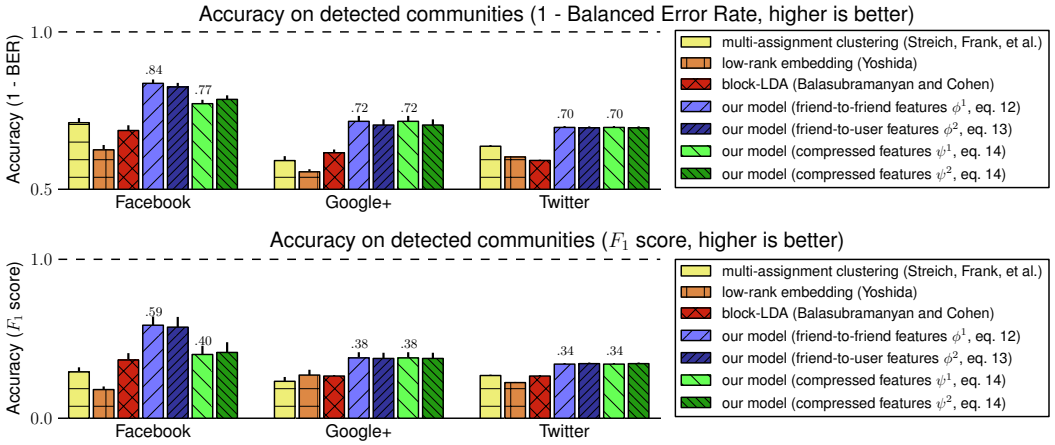

Figure 3: Performance on Facebook, Google+, and Twitter, in terms of the Balanced Error Rate (top), and the F_1 score (bottom). Higher is better. Error bars show standard error. The improvement of our best features ϕ^1 compared to the nearest competitor are significant at the 1% level or better.

Baselines. We considered a wide number of baseline methods, including those that consider only network structure, those that consider only profile information, and those that consider both. First we experimented with *Mixed Membership Stochastic Block Models* [2], which consider only network information, and variants that also consider text attributes [5, 6, 13]. For each node, mixed-membership models predict a stochastic vector encoding partial circle memberships, which we threshold to generate 'hard' assignments. We also considered *Block-LDA* [3], where we generate 'documents' by treating aspects of user profiles as words in a bag-of-words model.

Secondly, we experimented with classical clustering algorithms, such as *K-means* and *Hierarchical Clustering* [9], that form clusters based only on node profiles, but ignore the network. Conversely we considered *Link Clustering* [1] and *Clique Percolation* [21], which use network information, but ignore profiles. We also considered the *Low-Rank Embedding* approach of [30], where node attributes *and* edge information are projected into a feature space where classical clustering techniques can be applied. Finally we considered *Multi-Assignment Clustering* [23], which is promising in that it predicts hard assignments to multiple clusters, though it does so without using the network.

Of the eight baselines highlighted above we report the three whose overall performance was the best, namely Block-LDA [3] (which slightly outperformed mixed membership stochastic block models [2]), Low-Rank Embedding [30], and Multi-Assignment Clustering [23].

Performance on Facebook, Google+, and Twitter Data. Figure 3 shows results on our Facebook, Google+, and Twitter data. Circles were aligned as described in (eq. 15), with the number of circles $\hat{K}$ determined as described in Section 3. For non-probabilistic baselines, we chose $\hat{K}$ so as to maximize the *modularity*, as described in [20]. In terms of absolute performance our best model ϕ^1 achieves BER scores of 0.84 on Facebook, 0.72 on Google+ and 0.70 on Twitter (F_1 scores are 0.59, 0.38, and 0.34, respectively). The lower F_1 scores on Google+ and Twitter are explained by the fact that many circles have not been maintained since they were initially created: we achieve high recall (we recover the friends in each circle), but at low precision (we recover additional friends who appeared after the circle was created).

Comparing our method to baselines we notice that we outperform all baselines on all datasets by a statistically significant margin. Compared to the nearest competitors, our best performing features ϕ^1 improve on the BER by 43% on Facebook, 26% on Google+, and 16% on Twitter (improvements in terms of the F_1 score are similar). Regarding the performance of the baseline methods, we note that good performance seems to depend critically on predicting *hard* memberships to *multiple* circles, using a combination of *node and edge* information; none of the baselines exhibit precisely this combination, a shortcoming our model addresses.

Both of the features we propose (friend-to-friend features ϕ^1 and friend-to-user features ϕ^2) perform similarly, revealing that both schemes ultimately encode similar information, which is not surprising,

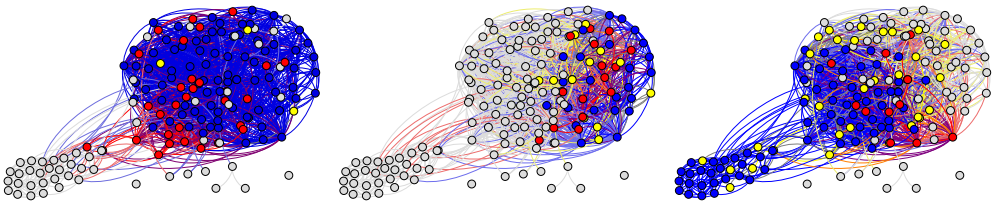

Figure 4: Three detected circles on a small ego-network from Facebook, compared to three ground-truth circles (BER $\simeq 0.81$). Blue nodes: true positives. Grey: true negatives. Red: false positives. Yellow: false negatives. Our method correctly identifies the largest circle (left), a sub-circle contained within it (center), and a third circle that significantly overlaps with it (right).

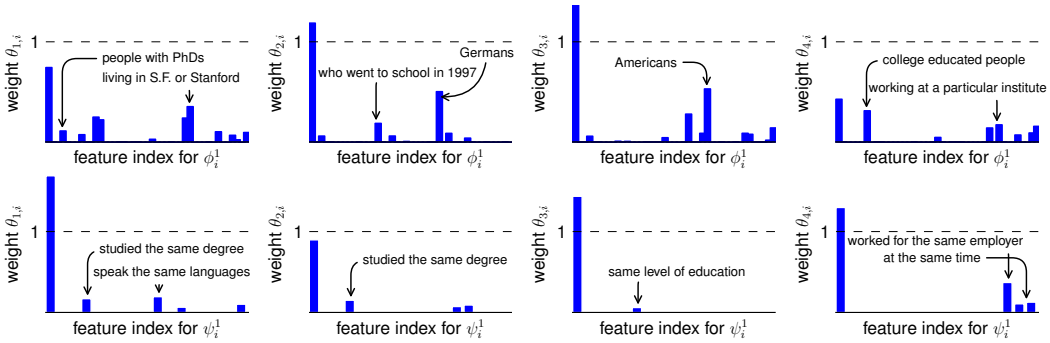

Figure 5: Parameter vectors of four communities for a particular Facebook user. The top four plots show 'complete' features ϕ^1, while the bottom four plots show 'compressed' features ψ^1 (in both cases, BER $\simeq 0.78$). For example the former features encode the fact that members of a particular community tend to speak German, while the latter features encode the fact that they speak the same language. (Personally identifiable annotations have been suppressed.)

since users and their friends have similar profiles. Using the 'compressed' features ψ^1 and ψ^2 does not significantly impact performance, which is promising since they have far lower dimension than the full features; what this reveals is that it is sufficient to model *categories* of attributes that users have in common (e.g. same school, same town), rather than the attribute values themselves.

We found that all algorithms perform significantly better on Facebook than on Google+ or Twitter. There are a few explanations: Firstly, our Facebook data is *complete*, in the sense that survey participants manually labeled *every* circle in their ego-networks, whereas in other datasets we only observe publicly-visible circles, which may not be up-to-date. Secondly, the 26 profile categories available from Facebook are more informative than the 6 categories from Google+, or the tweet-based profiles we build from Twitter. A more basic difference lies in the nature of the networks themselves: edges in Facebook encode *mutual* ties, whereas edges in Google+ and Twitter encode follower relationships, which changes the role that circles serve [27]. The latter two points explain why algorithms that use either edge or profile information in isolation are unlikely to perform well on this data.

Qualitative analysis. Finally we examine the output of our model in greater detail. Figure 4 shows results of our method on an example ego-network from Facebook. Different colors indicate true-, false- positives and negatives. Our method is correctly able to identify overlapping circles as well as sub-circles (circles within circles). Figure 5 shows parameter vectors learned for four circles for a particular Facebook user. Positive weights indicate properties that users in a particular circle have in common. Notice how the model naturally learns the social dimensions that lead to a social circle. Moreover, the first parameter that corresponds to a constant feature '1' has the highest weight; this reveals that membership to the same community provides the strongest signal that edges will form, while profile data provides a weaker (but still relevant) signal.

Acknowledgements. This research has been supported in part by NSF IIS-1016909, CNS-1010921, IIS-1159679, DARPA XDATA, DARPA GRAPHS, Albert Yu & Mary Bechmann Foundation, Boeing, Allyes, Samsung, Intel, Alfred P. Sloan Fellowship and the Microsoft Faculty Fellowship.

References

[1] Y.-Y. Ahn, J. Bagrow, and S. Lehmann. Link communities reveal multiscale complexity in networks. *Nature*, 2010.

[2] E. Airoldi, D. Blei, S. Fienberg, and E. Xing. Mixed membership stochastic blockmodels. *JMLR*, 2008.

[3] R. Balasubramanyan and W. Cohen. Block-LDA: Jointly modeling entity-annotated text and entity-entity links. In *SDM*, 2011.

[4] E. Boros and P. Hammer. Pseudo-boolean optimization. *Discrete Applied Mathematics*, 2002.

[5] J. Chang and D. Blei. Relational topic models for document networks. In *AIStats*, 2009.

[6] J. Chang, J. Boyd-Graber, and D. Blei. Connections between the lines: augmenting social networks with text. In *KDD*, 2009.

[7] Y. Chen and C. Lin. *Combining SVMs with various feature selection strategies*. Springer, 2006.

[8] M. Handcock, A. Raftery, and J. Tantrum. Model-based clustering for social networks. *Journal of the Royal Statistical Society Series A*, 2007.

[9] S. Johnson. Hierarchical clustering schemes. *Psychometrika*, 1967.

[10] D. Kim, Y. Jo, L.-C. Moon, and A. Oh. Analysis of twitter lists as a potential source for discovering latent characteristics of users. In *CHI*, 2010.

[11] P. Krivitsky, M. Handcock, A. Raftery, and P. Hoff. Representing degree distributions, clustering, and homophily in social networks with latent cluster random effects models. *Social Networks*, 2009.

[12] P. Lazarsfeld and R. Merton. Friendship as a social process: A substantive and methodological analysis. In *Freedom and Control in Modern Society*. 1954.

[13] Y. Liu, A. Niculescu-Mizil, and W. Gryc. Topic-link LDA: joint models of topic and author community. In *ICML*, 2009.

[14] D. MacKay. *Information Theory, Inference and Learning Algorithms*. Cambrdige University Press, 2003.

[15] J. McAuley and J. Leskovec. Discovering social circles in ego networks. arXiv:1210.8182, 2012.

[16] M. McPherson. An ecology of affiliation. *American Sociological Review*, 1983.

[17] A. Menon and C. Elkan. Link prediction via matrix factorization. In *ECML/PKDD*, 2011.

[18] A. Mislove, B. Viswanath, K. Gummadi, and P. Druschel. You are who you know: Inferring user profiles in online social networks. In *WSDM*, 2010.

[19] P. Nasirifard and C. Hayes. Tadvise: A twitter assistant based on twitter lists. In *SocInfo*, 2011.

[20] M. Newman. Modularity and community structure in networks. *PNAS*, 2006.

[21] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 2005.

[22] C. Rother, V. Kolmogorov, V. Lempitsky, and M. Szummer. Optimizing binary MRFs via extended roof duality. In *CVPR*, 2007.

[23] A. Streich, M. Frank, D. Basin, and J. Buhmann. Multi-assignment clustering for boolean data. *JMLR*, 2012.

[24] J. Ugander, B. Karrer, L. Backstrom, and C. Marlow. The anatomy of the Facebook social graph. preprint, 2011.

[25] C. Volinsky and A. Raftery. Bayesian information criterion for censored survival models. *Biometrics*, 2000.

[26] D. Vu, A. Asuncion, D. Hunter, and P. Smyth. Dynamic egocentric models for citation networks. In *ICML*, 2011.

[27] S. Wu, J. Hofman, W. Mason, and D. Watts. Who says what to whom on twitter. In *WWW*, 2011.

[28] J. Yang and J. Leskovec. Community-affiliation graph model for overlapping community detection. In *ICDM*, 2012.

[29] J. Yang and J. Leskovec. Defining and evaluating network communities based on ground-truth. In *ICDM*, 2012.

[30] T. Yoshida. Toward finding hidden communities based on user profiles. In *ICDM Workshops*, 2010.

[31] J. Zhao. Examining the evolution of networks based on lists in twitter. In *IMSAA*, 2011.