Automatic Grouping for Social Networks
CS229 Project Report

Xiaoying Tian
Statistics, Stanford University

Ya Le
Statistics, Stanford University

Yangru Fang
Statistics, Stanford University

Abstract

Social networking sites allow users to manually categorize their friends, but it is laborious to construct and keep updating those categories when a user's network grows. Leskovec et al. [2] defines an unsupervised model to identify a user's social circles. However, in real life users have personal preferences about how to group their friends. Indeed, it is possible that two users have exactly the same social networks but categorize their friends differently. In such case, unsupervised methods will fail to capture such personal preferences as they don't incorporate information about what kinds of social circles the user finds valuable. In this paper, we develop a supervised model for detecting social circles that combines network structure as well as user information. Experiments show that our model achieves significantly higher accuracy than K-means and Naive Bayes, and has comparable overall performance to that in Leskovec et al.'s work with lower computational complexity. Our method also turns out to have best performance on relatively small networks.

1. Introduction

As social network sites get bigger and more cluttered, categorizing friends into different social circles becomes a major mechanism for users to organize their social networks and cope with overwhelming volumes of information generated by their friends. Users in major social network sites (e.g. Google+, Facebook and Twitter) categorize their friends either manually or simply by grouping friends sharing a common attribute. The goal of our project is to set up a system which automatically categorizes a user's friends. We incorporate concepts from social network analysis into machine learning techniques to solve the above problem.

Research has been done on this topic via both conventional machine learning approaches such as decision trees (Baatarjav et al. [1]), and also social network techniques (Leskovec & McAuley [2]). Leskovec et al. [2] proposed an unsupervised method to tackle this problem. We propose a new model that uses this method as a component.

Given a single user, a network is formed by his/her friends. Following [2], we refer to this user as the ego and this network as its ego-network. In our project, we formulate this problem as a supervised learning problem and take into account both the profile information and the network structure.

Our method also differs from conventional clustering methods in the sense that the clusters can overlap with each other. We introduce a discriminative model to identify social circles based on the fact that circles tend to be densely connected with members sharing some common traits. With maximum likelihood estimation, our algorithm can learn the structure of the social circles as well as common features within each circle. Additionally, we compare our algorithm with both the K-means algorithm and Naive Bayes as baselines.

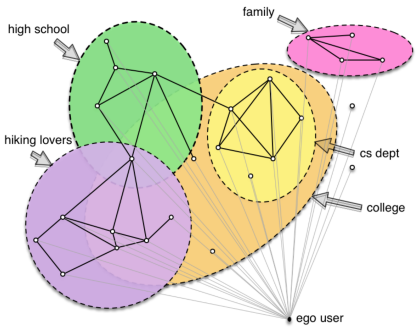

Figure 1: Sample circle diagram

2. Dataset Description

The dataset we used is the Facebook dataset in [2], which contains 9 ego networks comprised of 4039 users and an undirected social network with 88234 friendship connections. The profile information is collected in 26 categories, including languages, hometowns, birthdays, locations, etc. Social circle labels were obtained by asking the 9 egos to

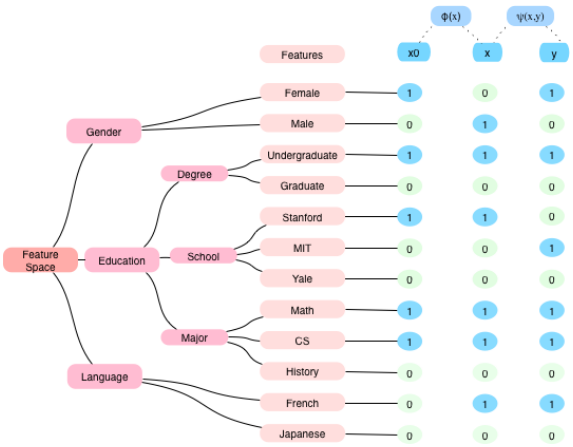

Figure 2: Feature space diagram

manually identify all the circles to which their friends belong. On average, there are 19 circles in each ego-network with an average size of 22 friends.

3. Feature Construction

The profile of a single user can be represented as a tree where each level encodes increasingly specific information. (See Figure 2). We construct the feature space by aggregating all the user attributes in a ego network and represent a single user's profile information as a binary vector, where '1' indicates the user has this attribute. For example (Figure 2), user x has profile [Gender: Male, Education: Degree: Undergrad, Education: School: Stanford, Education: Major: CS, Education: Major: Math, Language: French]. Then his profile vector is: [0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0]. Note that such profile vectors are defined per ego-network. For example, although thousands of companies exist in the whole Facebook network, only a few appear among any particular ego network.

Let $T^x = (T_1^x, \ldots, T_n^x)$ denote user x's profile vector. We define the *difference* vector $\sigma_l^{x,y} = \delta(T_l^x, T_l^y)$, as an indicator of whether the two users x and y differ at feature l. We define $s^{x,y} = 1 - \sigma^{x,y}$ as the *similarity* vector. Suppose the ego user is x_0. We construct the following two features, one associated with nodes and the other associated with edges: $\phi(x) = s^{x,x_0}$, the similarity vector between user x and the ego only and also $\psi(x, y) = s^{x,y}$, pairwise similarity vector between x and y.

4. Methodology

4.1. *K*-means

We use the *K*-means method as our unsupervised baseline model to detect social circles in the Facebook data. We implement the algorithm using the feature mapping $\phi(x)$

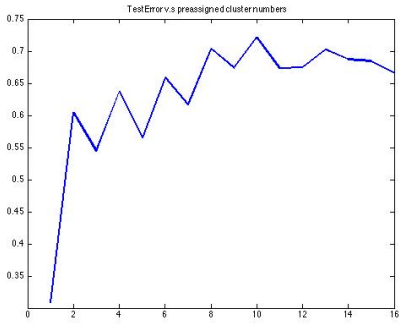

Figure 3: Test error v.s Number of preassigned centroids

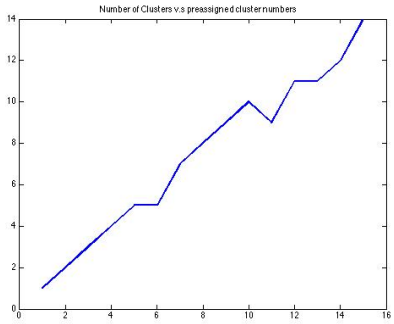

Figure 4: Number of resulting centroids v.s K

only and let the preassigned number of clusters range from 1 to 16. Figure 3 shows that as the preassigned clusters number increases, the test error increases as well. Notably, K-means works the best in the degenerate case where there is only one cluster. This indicates that K-means is not the right model for this problem and/or featurization.

Another issue with K-means is that the number of resulting centroids may be less than K because some of the clusters merge. In this problem, we observe this as we increase the parameter K. (See Figure 4) This indicates that a lot of the clusters formed are garbage clusters, and increasing K hurts accurate prediction. This situation occurs because: first, the feature vector for each user is sparse with binary outcome (as opposed to continuous outcome which is more appropriate for the K-means method) and second, in reality, the social circles in a social network do overlap, thus a clustering algorithm is not proper here.

4.2. Naive Bayes

We implement Naive Bayes with feature map $\phi(x)$ as our supervised baseline model. Recognizing that social circles may overlap, we encode the social circles to which a user x belongs as $c^x = (c_1^x, \ldots, c_K^x)$ where c_l^x is a binary variable indicating whether user x is in circle C_l, and K is the total

number of circles. For each circle C_l, we use c_l^x's as classification labels, and perform the Naive Bayes algorithm for this particular circle. In this way, we obtain K classifiers $(h_1, \ldots, h_K)$, with h_l denoting the classifier for circle C_l. The algorithm yields an average test error of 47.05%. The high test error is the result of some particularly big circles in the network; some circles cover up to 70% of the users. Naive Bayes is very likely to identify these circles while ignoring other smaller circles. In some extreme cases, the algorithm will assign users apparently at random to each circle according to their size in the training data, regardless of the user's feature vector.

4.3. Our model

In this section, we improve the featurization and propose a more sophisticated model to better solve the problem.

4.3.1 Featurization

1. **Feature Space Dimension Reduction**

   Both the previous two algorithms suffer from high-dimensional feature spaces. Noticing that similarity vectors are sparse and that each entry of the vectors corresponds to a leaf node in the profile tree (Figure 2), we address the issue by summing up the entries belonging to the same category. More specifically, $\tilde{s}_p^{x,y} = \sum_{l \in children(p)} s_l^{x,y}$, where p denotes category p. This achieves a reduction in feature space dimension from over 300 to 26 for the Facebook data.

2. **Network Structure**

   At this point, we have only used $\phi(x)$, the profile information for each user as our feature vectors. However, we would also like to take into account the similarity between the users to improve our model. More specifically, we will also incorporate the similarity vector $\psi(x, y)$ between two users x and y to explore the network structure. As members of the same social circle tend to be densely connected, this will provide important information about the social circle formation.

4.3.2 Proposed Model

We propose a discriminative model which considers both the profile information and the network structure in order to identify the social circles. The input to our model is an ego-network $G = (V, E)$, along with the feature vectors $\phi(x)$ and $\psi(x, y)$ and circle labels. V and E denote the node set and the edge set of the ego-network. Suppose the users are $\{x^{(1)}, \ldots, x^{(m)}\}$, with corresponding circle labels $\{c^{(1)}, \ldots, c^{(m)}\}$. We denote the feature vectors of all users as Φ and Ψ. For each circle C_l, let θ_l denote the parameter vector associated with shared features within the circle and let α_l denote some trade-off parameter which will be explained later. Our algorithm will yield θ_l, α_l by maximizing the following log-likelihood:

$$\begin{aligned} l(\theta, \alpha) &= \log(p(\mathcal{C}, G|\Phi, \Psi; \theta, \alpha)) \\ &= \log(p(\mathcal{C}|\Phi; \theta) p(G|\mathcal{C}, \Psi; \theta, \alpha)). \end{aligned} \tag{1}$$

The log-likelihood consists of two parts: the first part is the likelihood of the circle label $\mathcal{C}$ based only on the node features $\phi(x)$, and the second part is the likelihood of the edge set $\mathcal{E}$ based on the edge features $\psi(x, y)$ and the different circles $\mathcal{C}$. Since the circles C_l and the edges $e = (x, y)$ are generated independently, we will have:

$$\begin{aligned} l_1 &= \log\ p(\mathcal{C}|\Phi; \theta) \\ &= \log \prod_{i=1}^{m} p(c^{(i)}|\phi(x^{(i)}); \theta) \\ &= \sum_{i=1}^{m} \sum_{l=1}^{K} \log\ p(c_l^{(i)}|\phi(x^{(i)}); \theta_l) \end{aligned} \tag{2}$$

$$\begin{aligned} l_2 &= \log\ p(G|\mathcal{C}, \Psi; \theta, \alpha) \\ &= \log \prod_{e \in E} p(e \in E|\mathcal{C}, \Psi; \theta, \alpha) \prod_{e \notin E} p(e \notin E|\mathcal{C}, \Psi; \theta, \alpha) \\ &= \sum_{e \in E} \log\ p(e \in E|\mathcal{C}, \Psi; \theta, \alpha) \\ &\quad + \sum_{e \notin E} \log\ p(e \notin E|\mathcal{C}, \Psi; \theta, \alpha) \end{aligned} \tag{3}$$

We use the logistic regression model to form the likelihood of the circle labels, i.e., $p(c_l^{(i)} = 1|\phi(x^{(i)}); \theta_l) = g(\theta_l^T \phi(x^{(i)}))$, where g is the sigmoid function. For the likelihood of the edge set $\mathcal{E}$ in the graph, we observe that an edge between x and y is likely to form if they belong to the same circle C_l in which case $\theta_l^T \psi(x, y)$ tends to be high. [2]. Thus the probability of $e = (x, y) \in \mathcal{E}$ is:

$$\begin{aligned} &p(e \in E|\mathcal{C}, \psi(e); \theta, \alpha) \\ &\propto exp\{ \sum_{C_l : \{x,y\} \subseteq C_l} \theta_l^T \psi(e) - \sum_{C_l : \{x,y\} \not\subseteq C_l} \alpha_l \cdot \theta_l^T \psi(e) \} \end{aligned} \tag{4}$$

where α_l determines the amount we penalize if $x, y \notin C_l$. Also let:

$$d_l(e) = \delta(\{x, y\} \subseteq C_l) - \alpha_l \cdot \delta(\{x, y\} \not\subseteq C_l) \tag{5}$$

$$D(e) = \sum_{l=1}^{K} \theta_l^T \psi(e) d_l(e) \tag{6}$$

Then with the fact $p(e \in E) + p(e \notin E) = 1$, we got:

$$p(e \in E) = \frac{e^{D(e)}}{1 + e^{D(e)}} \qquad p(e \notin E) = \frac{1}{1 + e^{D(e)}} \tag{7}$$

By plugging eq. 7 into eq. 3, we get

$$l_2 = \sum_{e \in E} D(e) - \sum_{e \in V \times V} \log\left(1 + e^{D(e)}\right) \tag{8}$$

Both l_1 and l_2 are concave, thus we are able to optimize $l = l_1 + l_2$ through gradient ascent. The update rule goes as follows:

$$\frac{\partial l(\theta, \alpha)}{\partial \theta_l} = \sum_{i=1}^{m} [c_l^{(i)} - g(\theta_l^T \phi(x^{(i)}))] \phi(x^{(i)}) + \sum_{e \in E} d_l(e)\psi(e) - \sum_{e \in V \times V} \frac{e^{D(e)}}{1 + e^{D(e)}} d_l(e)\psi(e) \tag{9}$$

$$\frac{\partial l(\theta, \alpha)}{\partial \alpha_l} = \sum_{e \in E} -\theta_l^T \psi(e) \delta(\{x, y\} \not\subseteq C_l) + \sum_{e \in V \times V} \frac{e^{D(e)}}{1 + e^{D(e)}} \theta_l^T \psi(e) \delta(\{x, y\} \not\subseteq C_l) \tag{10}$$

We randomly select 70% of the users in an ego-network as our training data and obtain $\theta_l's$ and $\alpha_l's$ by maximizing eq.1 using the gradient ascent update rules defined above. To predict the circle labels of some user x_i in the test dataset, we compute the likelihood of $p(x_i \in C_l, \tilde{G} | \Theta, \Phi, \Psi)$ for each circle C_l, where $\tilde{G}$ is the new network after adding x_i. Then x_i is predicted to belong to the top J circles that have the largest likelihoods. Our results show that $J = 3$ usually gives very good predictions, while one can also select J via cross-validation.

4.4. Evaluation Metrics

We evaluate our method by examining the differences between the circles our algorithm selects $\hat{\mathcal{C}} = \{\hat{C}_1, \ldots, \hat{C}_{\hat{K}}\}$ and the true circle labels $\mathcal{C} = \{C_1, \ldots, C_K\}$. We adopt the Balanced Error Rate (BER) as a difference measure between the two circles [3], and take the average BER of all the circles as our error rate.

$$BER(\hat{C}, C) = \frac{1}{2}\left(\frac{|\hat{C} \backslash C|}{|\hat{C}|} + \frac{|C \backslash \hat{C}|}{|C|}\right). \tag{11}$$

For unsupervised learning methods like the K-means algorithm, we don't know the correspondence between the circles in $\hat{\mathcal{C}}$ and $\mathcal{C}$. As a matching heuristic, we align the circles of these two types by minimizing

$$f(i) = argmin_j(||\hat{\mu}_i - \mu_j||_2), \tag{12}$$

where $\hat{\mu}_i$ and μ_i are the centroids of $\hat{C}_i$ and C_i respectively. Therefore f defines a correspondence between $\hat{\mathcal{C}}$ and $\mathcal{C}$, i.e., $C_{f(i)}$ is the corresponding circle for $\hat{C}_i$.

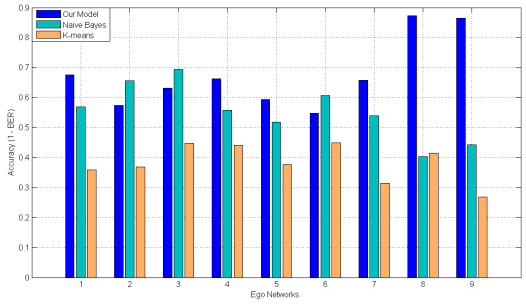

Figure 5: Accuracy comparison

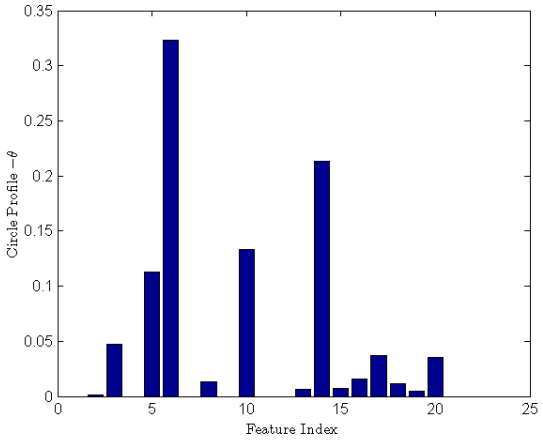

Figure 6: $-\theta_i$ for circle 6 in ego-network 1

5. Experiment & Results

During the implementation, we anneal the learning rate α to accelerate the learning speed. The comparison of the three methods we implemented is shown in Figure 5. As expected, we observe that the K-means method performs the worst, and our method outperforms the Naive Bayes method for 6 ego-networks out of 9.

Figure 6 plots the parameter vector $-\theta_6$ for circle C_6 in ego-network 1. The $5^{th}, 6^{th}, 10^{th}$ and 14^{th} entries in the vector are significantly larger than the other entries. We further examine the corresponding categories in ego-network 1 and find that those entries correspond to *Education: School*, *Education: Type*, *Gender* and *Locale* (i.e. Location), which are important features for social network detection. We also plot the prediction results of a circle on ego-network 3 as in Figure 7. In the plot, densely connected nodes form a cluster. The result shows that our model successfully detects almost all the members of the circle.

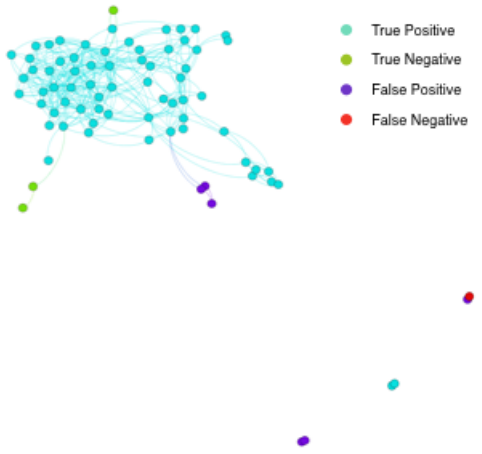

Figure 7: Prediction graph on ego-network 3

6. Conclusion and Future Work

We introduce a way of combining the user profile information and the social network structure to detect the social circles to which a user belongs in an ego-network. As a supervised model, our method captures ego users' personal preferences in grouping their friends, and it also outperforms the methods which only consider the user profile information. Also, it is reasonably common that users in the same social circle are also friends with each other, which will result in interesting graph structures that we can take advantage of in circle detection. For prediction we now pick the top J circles of the highest probabilities as the circles a user belongs to. In order to improve the model, we can use cross validation to decide the number of the circles each user belongs to. Also we can boost the efficiency of the algorithm by eliminating the irrelevant features in feature space reduction.

References

[1] E. Baatarjav, S. Phithakkinukoon and R. Dantu. On the Move to Meaningful Internet Systems. *OTM 2008 Workshops Lecture Notes in computer Science*. Vol. 5333, pp. 211-219, 2008.

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

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

[4] J. Friedman. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 2002.

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

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

[7] T. Hastie, R. Tibshirani, and J. Friedman. The Elements of Statistical Learning. *Springer Series in Statistics Springer New York Inc., New York, NY, USA*, 2001

[8] J. A. Hartigan and M. A. Wong. A K-Means Clustering Algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, Vol. 28, No. 1, pp. 100-108, 1979.