# Social Network Circle Discovery Using Latent Dirichlet Allocation

Frank Fan; Jaimie Xie; Matthew Kim

## ABSTRACT

Online Social Networks, such as Facebook, provide a great interface for connecting with others, whether they are acquaintances or close friends. However, there is no distinction made between different *social circles*, which are clusters of friends who share some common feature(s). In this paper, we explore ways to apply *Latent Dirichlet Allocation (LDA)*, an unsupervised learning algorithm traditionally used for topic detection in textual corpora, to automatically detect social circles among a subject's friends. For each friend, which we will consider as *documents*, we take in account both the profile features and users' friends, comparable to word "tokens." Finally, we will analyze our results by finding the cost-minimizing assignment from our circles to the ground-truth circles, based on the *Balanced Error Rate (BER)*.

## CONTACT

Jaimie Xie
Email: jaimiex@stanford.edu
Phone: (678) 761-3223
Frank Fan
Email: ffan9@stanford.edu
Phone: (650) 395-8299
Matthew Kim
Email: mdkim@stanford.edu
Phone: (608) 630-1980

## BACKGROUND

**Latent Dirichlet Allocation**
LDA is a generative algorithm traditionally used in NLP for document topic-modeling [4]. LDA models documents as mixtures of *topics*, with each *topic* being a multinomial distribution of words. In effect, the probability of a document is:

$$p(\mathbf{w}|\alpha,\beta) = \frac{\Gamma\sum_i \alpha_i}{\prod_i \Gamma(\alpha_i)} \times \int \left(\prod_{i=1}^{k} \theta_i^{\alpha_i - 1}\right) \left(\prod_{n=1}^{N} \sum_{i=1}^{k} \prod_{j=1}^{V} (\theta_i \beta_{ij})^{w_n^j}\right) d\theta$$

We extend this algorithm to apply it to social circle discovery -- circles representing topics, users representing documents and user features representing words. Not only is this an intuitive extension of the field of linguistic topic-modeling, LDA proves to be more time-efficient than traditional circle-discovery methods [1]

In addition, in line with work by Hoffman et al. [3], we will utilize an online-learning variant of LDA [6] for this project.

## EXPERIMENT

**Modeling the likelihood of the dataset:**
Given a set of users, an LDA model produces a metric called perplexity, related to the log-likelihood as [4]:

$$Perplexity(D) = \exp\left(\frac{-LL(D)}{N}\right)$$

where D is a set of users and N is the total number of features, summed across all users in D.

**Selection of the Number of Circles, with $AIC_c$:**
We select the number of circles through stepwise selection that minimizes the $AIC_c$ criterion: [1][2]

$$AIC_c = -2LL + 2p + \frac{2p(p+1)}{nA - p - 1}$$

which rewards high likelihood, but penalizes model complexity, and corrects for small sample sizes relative to the dimension of the model's parameter space.

**Unsupervised learning**
We then run online-LDA [3] using the selected circle number to model the structure of the dataset. Let us call this algorithm **LDA+C.**

For comparison, using the same $AIC_c$-selected circle number above, we also explored a k-means clustering algorithm. For this, we preprocessed the data using a truncated SVD representation [5] to reduce dimensionality in terms of the number of features. We will refer to this as **KMEANS+C**

Finally, for comparison/evaluation, we ran the above algorithms using the ground-truth number of circles of the networks. Let us call these algorithms **LDA** and **KMEANS**.

## RESULTS/ANALYSIS

After running the LDA Algorithm, we choose a cut-off probability to choose which circles each user should actually be assigned to. We placed user *u* in circle C if
$\Pr(u \in C) > 1/N$, where *N* is the number of circles we predicted. After establishing the circles, we compare the circles with the *Ground-truth circles*. For our experiment, we used the Balanced Error Rate (BER) as the error/cost function to minimize total error of mapping circles, as did Petkos et al (2015). If we let $C = \{C_1, C_2, ...C_k\}$ be the set of automatically produced circles, and $D = \{D_1, D_2, ...D_k\}$ be the set of ground-truth. Then, we can define the BER as:

$$BER(C_i, D_i) = \frac{1}{2}\left(\frac{|D_i \backslash C_i|}{|D_i|} + \frac{|D_i^c \backslash C_i^c|}{|D_i^c|}\right)$$

The BER cost function equally weights the fraction false-positives and false-negatives. If we compute the BER for every pair $(C_i, D_j)$, we can construct the cost matrix where the *ij*-th entry is $BER(C_i, D_j)$.
We want to find a circle matching *f*: $C \mapsto D$, which gives us the least total error. With the Kuhn-Munkres algorithm, we can solve the assignment problem in $O(n^3)$ time instead of O(n!) by trying every possible *f*.
For our final *BER score*, we take the average of the BER rates from each circle assignment, then subtract that from one:

$$BER_f = \frac{1}{|f|} \sum_{C \in dom(f)} (1 - BER(C, f(C)))$$

For each algorithm we average the $BER_f$ values over all the ego networks. See Figure 1.

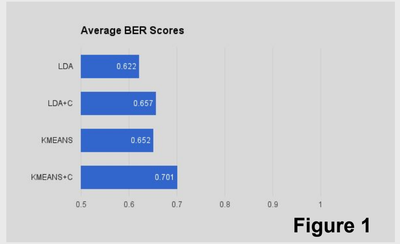


**Figure 1**

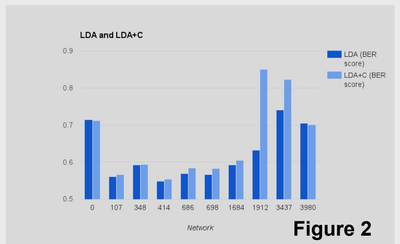


**Figure 2**

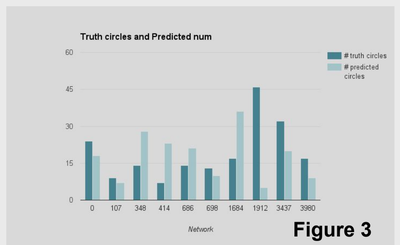


**Figure 3**

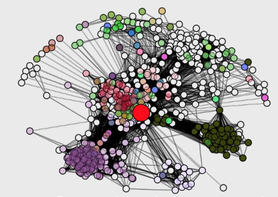

**Figure 4.** Ground truth on network 1912

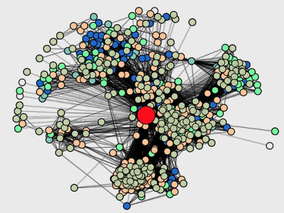

**Figure 5.** LDA on network 1912

**Figure 6.** KMeans on network 1912

## DATA[1]

The data is divided into *ego networks*, which consists of the *ego* node, all of the nodes it is connected to (called *alters*), and all of the edges there may be among these nodes. Within each ego network, we have the following:

- Circles: these are the circles that the user manually chose, the *ground-truth circle*.
- Edges: this contains every edge in the ego network, other than the implicit edges that connect each *alter* to the *ego node*.
- Features: for the ego and each alter, we are given a binary array, where a 1 in index i signifies that feature i is satisfied (and 0 otherwise). The features are constructed in a tree-structure:


- Feature names: this contains the names of the features that correspond with the feature arrays. In general, we will just use the numerical labeling of the features.

[1]The data that we used for training/testing is provided by the Stanford Network Analysis Project, and all of our data comes from Facebook.

## CONCLUSIONS

For both our K-means and LDA algorithms, we achieved better results when we predict the number of circles using $AIC_C$ , rather than just setting k = the number of ground-truth circles (See Figure 3). This is because in the latter case, we are overfitting the data. Predicting our own circles protected against model complexity, as in the 1912 ego-network.

Another surprising result was that the K-Means algorithms, which used only the feature vectors of the user's friends, but did not consider the network structure or user's own profile features, did better than the LDA algorithms, which considered all three components. Our implementation of the LDA algorithm places larger weight on the network structure because the user's friends "documents" are largely comprised of their connections within the ego-network, rather than profile features. This implies that profile features (even using the compressed vectors) are more powerful than network structure in informing us about circle formations.

## REFERENCES

1. Georgios Petkos, Symeon Papadopoulos, and Yiannis Kompatsiaris. 2015. Social Circle Discovery in Ego-Networks by Mining the Latent Structure of User Connections and Profile Attributes. In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015* (ASONAM '15), Jian Pei, Fabrizio Silvestri, and Jie Tang (Eds.). ACM, New York, NY, USA, 880-887.
2. C. Hurvich and C. Tsai, Regression and time series model selection in small samples, Biometrika 76, 297307, 1989.
3. Hoffman, M., Bach, F. R., & Blei, D. M. (2010). Online learning for latent dirichlet allocation. In *advances in neural information processing systems* (pp. 856-864).
4. Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *the Journal of machine Learning research*, *3*, 993-1022.
5. Berry, M. W., Dumais, S. T., & O'Brien, G. W. (1995). Using linear algebra for intelligent information retrieval. *SIAM review*, *37*(4), 573-595.
6. Radim Rahurek and Peter Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. ELRA