# On Group Popularity Prediction in Event-Based Social Networks


## ABSTRACT

Event-based social networks have recently emerged as an important complement to online social networks. They enjoy the advantages of both online social networks and offline social communities: offline social events can be conveniently organized online, and users interact with each other face-to-face in the organized offline events. Although previous work has shown that member and structural features are important to the future popularity of the groups, it is not yet clear how different member roles and the interplay between them contribute to group popularity. In this paper, we study a real-world dataset from Meetup — a popular event-based social network platform — and propose a deep neural network based method to predict the popularity of new Meetup groups. Our method uses group-level features specific to event-based social networks, such as time and location of events in a group, as well as structural features internal to a group, such as the estimated member roles in a group and social substructures among members. Empirically, our approach reduces the RMSE of the popularity (measured in RSVPs) of a group's future events by up to 12%, against the state-of-the-art baselines. Moreover, through case studies, our method also identifies member and structure patterns that are most predictive of a group's future popularity. Our study provides new understanding about what makes a group successful in event-based social networks.



## CCS CONCEPTS

• **Information systems → Social networking sites**;

## KEYWORDS

Event-based Social Networks, Group Popularity Prediction, Circular Fingerprints, Role Discovery

**ACM Reference Format:**

. 2018. On Group Popularity Prediction in Event-Based Social Networks . In *Proceedings of The International World Wide Web Conference, Lyon, France, April. 2018 (WWW'18),* 10 pages.
https://doi.org/10.475/123_4


## 1 INTRODUCTION

As online social networks become more prevalent, people's face-to-face interactions are reshaped by these networks. In this work we focus on *event-based social network* (EBSN), an online social network whose members hold in-person events. Meetup [14] is one such online platform that allows members to find and join online interest groups, and organize face-to-face events in different categories, such as politics, books, games, movies, health, pets, careers, and hobbies, etc. While it is relatively easy to establish new groups in EBSN, it takes much more effort from the group organizers and members to make a group popular and sustainable. It is therefore important to understand the key factors contributing to the popularity and sustainability of groups in EBSN, especially newly established groups. Insights obtained from such a study can be used to guide the promotion, recommendation and investment on EBSN groups by EBSN platforms and investors.

*In this paper, we study the problem of group popularity prediction in EBSN, with a special focus on new groups.* More specifically, we focus on predicting the popularity (measured in number of RSVPs) of newly established interest groups in Meetup [14]. The main questions that we want to answer are: *1). can we predict the future success of new groups? 2). what are the observable factors that best predict a group's success?* Different from the previous studies on group popularity in the traditional online social networks, our study takes into account the unique features of EBSN in the following key aspects:

- *Get-out-of-the-couch Effort:* To participate in an offline social event, a user must be physically present at some specific venue at a scheduled time. Clearly, this takes more effort and commitment than participating in a pure online event. As a result, a user's attention (the ability to be active in multiple groups) is a severely limited resource, which must be accounted for in predicting the successes of competing groups.
- *Face-to-face Social Interactions with Implicit Social Relationships:* Once users meet in person, they may form stronger bounds than online interactions. Thus, to predict a group's success one must account for stronger and more sustainable social ties than online communities. However, EBSNs normally don't have the explicit social relations between group members, that are readily available in online social networks, such as "friends-with" (Facebook) and "follower" (Twitter).

*Contributions. We develop a novel approach to predict the popularity of newly formed groups in EBSN, achieving the state-of-the-art accuracy.* Our approach considers various factors: i) the group-level features, such as the past popularity of the group and the number of events, ii) the event-based features, such as location and schedule of events, iii) user-level features related to user's attention: how active is a user in an individual group, and how does the user distribute her activity/attention among multiple groups. Based on these features, the first key idea of our approach is the use of role discovery to determine the importance of users in a group and the roles they play. Similar to any social community in real life, a group in EBSN is more than a simple collection of members. A group's characteristics are mostly determined by the interaction among group members, and its success ultimately hinges on the "*chemistry*" among its members. Coincidentally, the second key aspect of our approach is the extension of circular and neural fingerprints techniques developed in chemistry [4, 22] to study how social ties between different types of users contribute to group success. Specifically, armed with each



*WWW'18, April. 2018, Lyon, France*



ACM ISBN 123-4567-24-567/08/06.
https://doi.org/10.475/123_4

group member's role, and event co-participation graphs generated from those members' activities, we combine the members' roles with these activity networks to predict a group's success, extending the fingerprints techniques developed to correlate the characteristics of atoms along with the neighboring bonds to other atoms to determine a molecule's function. Our extension also accounts for a user's limited attention by incorporating "attention-based" features, such as how many groups a user joined and how much time the user spent in each group into the member-level feature set.

By applying our novel approach to the Meetup dataset, we obtain interesting findings w.r.t. features that predict a group's success: (1) The user roles we discover in a group's social network are good predictors of the group's popularity, more than any other member-level features. (2) The most relevant user roles contributing to a group's popularity are not "organizer-like members", but "ordinary members" who have similar activity levels with their friends. (3) The most important substructures (interaction patterns) are not combining all the most important roles, but follow different combination patterns for different types of groups.

*Outline.* The rest of this paper is organized as follows. Section 2 introduces the related work on group behavior prediction in social networks, role discovery techniques and structural feature extraction. Section 3 describes the method we propose to solve the problem of predicting the groups' future popularity. In Section 4, we evaluate the performance of the proposed method using the Meetup dataset. Finally we conclude our work in Section 5.

## 2 RELATED WORK

**Event-based Social Networks**. Event-based social networks were first studied by Liu et al. [11]. Topic, location, and time preferences of individual users have been used to make event recommendations to users [1, 2, 8, 13, 18, 25, 26, 28]. In a related work, Liu et al. [12] and Pramanik et al. [17] have investigated group-level factors that contribute to a group's success or failure. Our work also focuses on predicting group popularity. We not only use a very different (and more effective) methodology, but also expand the feature set to include unique group-level features in EBSN, such as event venue and schedule, and combine member roles with the structure of social network.

**Group Popularity Prediction**. There are two major lines of research for this problem. One focuses on characterizing the evolution of online social network popularity by applying mean-field epidemic models to the time series of the "daily active user", without user-level or network structural information [20, 27]. The other focuses on using general group features to make predictions [12, 17, 19]. In contrast, our approach uses richer information and convolutes member roles, member's attention-capacity features, with their activity network structure to improve the prediction accuracy.

**Role Discovery**. The goal of network role discovery is to classify network nodes according to the roles they play in the network [5, 23, 24, 29]. Given a graph along with node features, the process of role discovery relies on defining node equivalence. Various types of equivalences have been introduced in previous studies, such as graph-based equivalences, feature-based equivalences, and hybrid equivalences [23]. To capture both structural and feature equivalences between members, the information that we use for role discovery represents both members' intrinsic behaviors, such as the numbers of events/groups that they have participated in, as well as the partial structural behaviors of members by including their one-hop-neighbors' features (see Table 1).

**Structural Feature Extraction**. Several studies on group popularity prediction have found that a group's main characteristics can be largely related to the interaction patterns among their members [12, 17, 19]. These patterns can be represented by the node attributes and link structures of a graph generated based on members' interactions. While multiple methods, such as "deepwalk" [16] and "node2vec" [6] have been proposed for mapping structured data to real-valued feature vectors, these mainly focus on transforming graphs into node features, instead of obtaining features for the graphs. Some recent studies [3, 9, 15, 21, 30] also focus on transforming substructures or subgraphs in large graphs to feature vectors, however the resulting latent vectors cannot be easily interpreted to obtain insights about group popularity. In contrast, we extend the circular and neural fingerprints techniques in chemistry informatics to gain important understanding on how subgraphs among different types of members contribute to group success.

## 3 PROPOSED METHOD

In this section, we first give a metric of a group's popularity and define our prediction problem in the context of the Meetup EBSN. We then propose our prediction method, leveraging on the group-level features (Section 3.2) and member-level features (Section 3.3). Finally, Section 3.4 presents our overall method combining these group-level and member-level features to make predictions.

### 3.1 Meetup Group Popularity Prediction

Meetup is an event-based social network in which users can form and join different interest groups online, and organize and participate in face-to-face social events offline. The group organizers create events, and each event has specified time, location and topic. The information about new events will be sent to group members through emails or website notifications. Each group member decides whether she will participate in the new events based on her time, location, and topic preferences, and then responds by sending RSVPs ("yes", "no", or "maybe"). Figure 1 illustrates the main components in Meetup social network. With the definitions of groups, events and users in Meetup, the group popularity prediction problem can be defined as:

*Definition 3.1.* Given all the related information of a group within a time window of $[0, n]$ months, and a time interval of $m$ months, predict the group's total RSVP number (popularity) within a future time window of $[n + m, 2n + m]$ months.

The time interval of $m$ months can be chosen to eliminate the effect of seasonal event holding patterns. For example, a skiing group may have high activity level only in winter (October ~ December) while the RSVP numbers in summer could be small. Using winter RSVP numbers to predict summer RSVP numbers, or vice versa, is not a meaningful prediction task. In our study, we calculate prediction accuracy for multiple choices of $m$, then take the average to represent the overall accuracy.

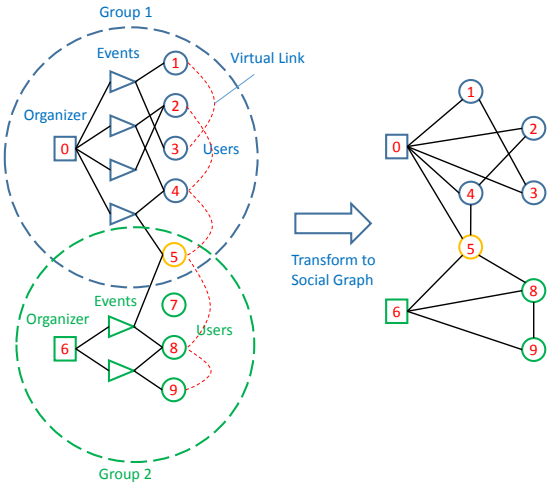


**Figure 1: Components in Meetup Social Network. The left-hand side shows components of two groups: every solid line from a user to an event represents a "yes" RSVP; When two users participate in the same events, a "virtual link" between them is created to represent their potential interaction; User "5" is the "social spanner" who joins two groups and user "7" is an inactive user who hasn't participated in any event; The right-hand side: using virtual links and removing events we get a homogeneous graph in which all nodes are users.**

## 3.2 Group-level Features

The most straight-forward features to use for popularity prediction are the summary statistics of each group. So we start with extracting "group-level features", which are various summary statistics of a group without examining the detailed features of each member in the group. We list the descriptions of fourteen group-level features for each Meetup group in Table 1, such as the scheduled time distributions of its events, the location distributions over its venues, and RSVP counts of all members, etc.

## 3.3 Internal Group Features

Although by only using group-level features we can achieve good performance, we intend to improve the accuracy further by investigating the internal group features. Internal features of a group can be defined as all the features that are related to each individual member in the group. These features should include the first-order features that can be directly calculated using basic statistics, such as the past attendance of a member and how many groups a member has joined. They should also include the second-order features that require further processing, such as member's role discovery and structural feature extraction.

*3.3.1 Member-level Feature Extraction.* We start with constructing social graph for each group from which the features are extracted. For each given group $g$, the social graph is defined as:

*Definition 3.2.* $G^g = (U^g, E^g, W^g)$, where $U^g$ is the member set in group $g$, $E^g$ denotes all edges between members and $W^g$ represents all edge weights. Two members $u_i$ and $u_j$ are defined to be connected if they co-participated in at least one event in group $g$. The weight on each edge is calculated as the number of events two users have co-participated in (Figure 1).

Following [7], we propose twelve member-level features listed in Table 1: Feature m1~m6 represent "who you are", i.e., features related to the member's own characteristics, and feature m7~m12 represent "who you know", i.e., features related to her neighbors' characteristics.

*3.3.2 Member Role Discovery.* To find role features of each group member, following [7], we use Non-negative Matrix Factorization (NMF) [10] for our role assignments. For a given a member-feature matrix $\mathbf{X} \in \mathbb{R}^{n \times f}$, we generate a rank-r approximation ($r < min(n, f)$) $\mathbf{MF} \approx \mathbf{X}$ where each row of $\mathbf{M} \in \mathbb{R}^{n \times r}$ represents a node's membership in each role and each column of $\mathbf{F} \in \mathbb{R}^{r \times f}$ represents how membership of a specific role contributes to the estimated feature values. With a distance measure $\|\cdot\|$, the problem can be simplified as:

$$\min_{\mathbf{F} \in \mathbb{R}^{r \times f}, \mathbf{M} \in \mathbb{R}^{n \times r}} \|\mathbf{MF\text{-}X}\|$$

subject to $\mathbf{M}, \mathbf{F} \geq 0$. In practice, considering the feature number $f = 12$, we choose the role number $r = 6$.

Taking the member-role matrix $\mathbf{M^g}$ for group $g$ generated by the role discovery method, we sum over all the members (rows) and get the group's role distribution vector $\left\{\mathbf{V_j^g} = \sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{M_{ij}^g}, \mathbf{1 \leq j \leq r}\right\}$, then we stack all the vectors $\{\mathbf{V^g}\}$ to form a group-role matrix $\mathbf{\Omega} \in \mathbb{R}^{\mathbf{p \times r}}$ where p represents the number of groups and r is the number of roles. Thus the correlation between role X and group popularity can be calculated as the correlation between the column corresponding to role X in $\mathbf{\Omega}$ and all groups' RSVP numbers.

In Figure 2 (top) we calculate the Pearson correlation between the member feature vectors and their group's popularity. The weak correlations (in the range of $[-0.2, +0.2]$) indicate that no individual member feature plays an important role in the popularity of group.

Figure 2 (bottom) shows that role discovery (i.e. combining multiple features to form a role) significantly increases the correlation, where roles are now positively strongly correlated with group popularity (in the range of $[+0.45, +0.75]$). Each role is a linear combination of multiple raw member features. Thus by analyzing the roles, we can get a better understanding of how certain combination of features contributes to the group's popularity, and at the same time, creates a role profile for each member.

*3.3.3 Structural Features.* As observed in the previous studies, structural features of a group play important roles in affecting the group's characteristics. However, the existing work does not consider the roles of members within the group when detecting structural patterns, while intuitively the members' roles should be taken into account. Let's consider an example group, "NY Tech", the largest group in the Meetup social network. Every week the organizers of "NY Tech" create a technical conference-like event and sometimes invite an expert on some topic to be the speaker (Figure 3). In this case, without differentiating between the two roles, "organizer" and "speaker", there is no difference between the two social network graphs. However, with the knowledge of their roles, one may easily notice that the subgraph surrounding

| | | |
|---|---|---|
| Group-level features | g1. Entropy of the time distribution over all times the events are held<br>g2. Average distance between any two events the group held<br>g3. Variance of the "event-event" distances<br>g4. Average distance between any event and any participating members<br>g5. Variance of the "event-member" distances<br>g6. Average distance between any member and any other member in the same group<br>g7. Variance of the "event-member" distances<br>g8. Entropy of the location distribution over all venues the event are held<br>g9. Density of the group's social graph<br>g10. Total degree of the group's social graph<br>g11. Event number the group has held<br>g12. Average RSVP number of all past events<br>g13. Variance of RSVP numbers of past events<br>g14. Sum RSVP number of past events. | Member-level features | m1. Total degree of the member<br>m2. Event number the member has participated in current group<br>m3. Group number the member has joined<br>m4. Entropy of member's attendance distribution over the groups the member joined<br>m5. Entropy of event number distribution over the groups the member joined<br>m6. Entropy of event fraction distribution over the groups the member joined<br>m7. Average degree of the member's 1-hop neighbors<br>m8. Average event number the "1-hop neighbors" have participated<br>m9. Average group number the "1-hop neighbors" have joined<br>m10. Average entropy of "1-hop neighbors'" attendance distribution over the groups they have joined<br>m11. Average entropy of "1-hop neighbors'" event number distribution over the groups they joined<br>m12. Average entropy of "1-hop neighbors'" event fraction distribution over the groups they joined |

**Table 1: Group-level and Member-level Features**

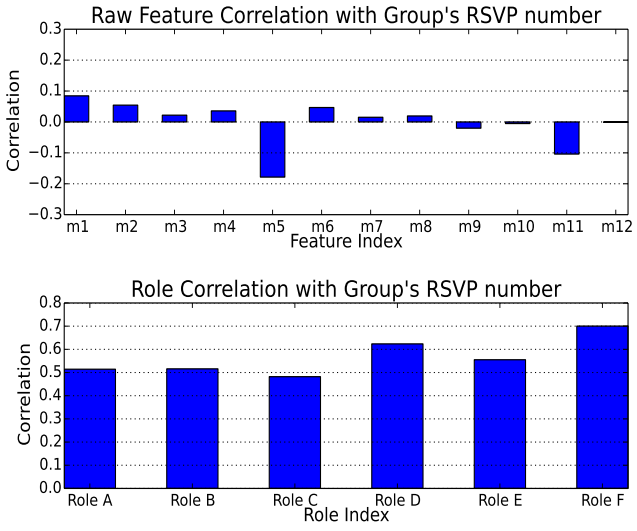


**Figure 2: Feature Correlation with Popularity. Role discovery method combines multiple raw member features into one distribution to represent a member's profile.**

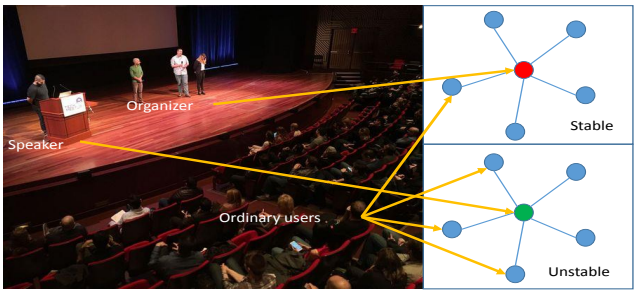


**Figure 3: A subgraph in the "NY Tech Meetup" Group**

the organizers (in the upper right) is more stable than the one surrounding the speaker (in the lower right) since the "speaker" is very likely to be changed in the next event, while the group's organizer remains the same.

**Circular Fingerprints.** In order to extract structural features embedded with nodes' roles, we use the "circular fingerprints" algorithm [22]. Circular fingerprints is a popular tool for handling graph-structured data in chemistry. It was first designed for molecular characterization, similarity searching, and structure-activity modeling. A molecule consists of atoms with different types. How different types of atoms are bounded together is the key factor that determines the characteristics of the molecule. In the context of event-based social network, we draw the analogy between a molecule and a group. We assume the members of a group are analogous to the atoms of a molecule, and the social ties between members are analogous to the chemical bonds (Figure 4). Then we can study how the subgraphs between members contribute to the group popularity using the circular fingerprints framework. The fingerprint generation process is accomplished mainly in two steps:

(1) The algorithm starts with assigning an initial identifier to each atom (member) in the molecule (group). This identifier captures some basic information of the atom (member) such as atomic number, connection count, etc. In our case the "basic information" is the role distribution attached to each member.
(2) After that, a number of iterations are performed to combine the initial atom (member) identifiers with identifiers of neighboring atoms (members) until a specified radius (number of hops from this atom) is reached. For example, in iteration 1, the identifiers of all "one-hop neighbors" of the target atom are combined with the identifier of the target atom to generate

the new identifier. Each iteration captures larger and larger circular neighborhoods around each atom (member), which are then encoded into single integer values using a suitable hashing method, and these identifiers are collected into a list. In this way, each subgraph is generated by a member along with her neighbors within a certain radius.

The identifier list (also called "fingerprints") is then used to characterize the properties of the molecule. In our case, we use the fingerprints as structural features to predict the group's popularity

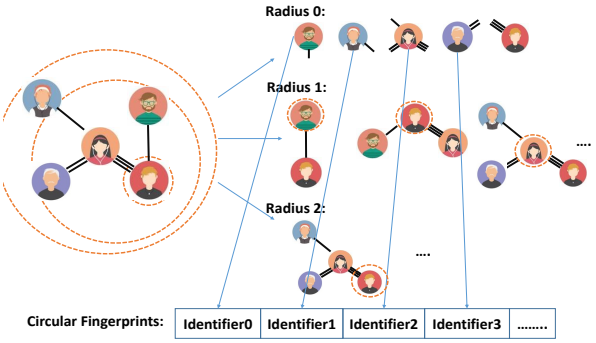


**Figure 4: Subgraphs Detected by Circular Fingerprints in Social Network. Circular Fingerprint scans the network for all subgraphs under certain radius. Each subgraph is then encoded into an integer identifier. Integer identifiers of all subgraphs constitute the circular fingerprints.**

**Group-Role Neural Fingerprints.** Although circular fingerprints is a convenient tool to study social graph, it has several limitations: 1) the algorithm can only handle graphs with fixed sizes; 2) even if the graphs vary a little bit, the resulting fingerprints can be quite different, making the features highly vulnerable to noise. Overcoming these limitations, Duvenaud et al. [4] proposed a convolutional neural network, where each neural network layer simulates the updating operation in circular fingerprints (Figure 5).

We now extend the convolutional fingerprinting algorithm to solve our problem. We denote this approach as the *Group-Role Neural Fingerprints*: For radius 0, the first hidden layer in the network takes the initial member-role matrix which is produced by the previous role discovery step as the input, then the output of this layer goes in two directions: in one direction the output is directly calculated as the radius 0 fingerprint; in the other direction, the output is updated with the adjacency matrix through a "feature update" operation. In this update operation, the member-role matrix is updated so that each member's 1-hop neighbors' role distribution vectors are added to the corresponding row of member-role matrix. In this way, the algorithm iterates until a certain radius is reached. After each iteration, more and more local structural information are captured. The detailed operations are presented in Algorithm 1. As the result, the final fingerprints are calculated as the summation of the fingerprints at each radius and then taken as input of linear regression algorithm to produce prediction of the RSVP number (popularity). The number of hidden layers equals to the given radius $R$. The neural network's weights $\mathbf{H^0}, ...\mathbf{H^R}$ and $W_0, ...W_R$ are

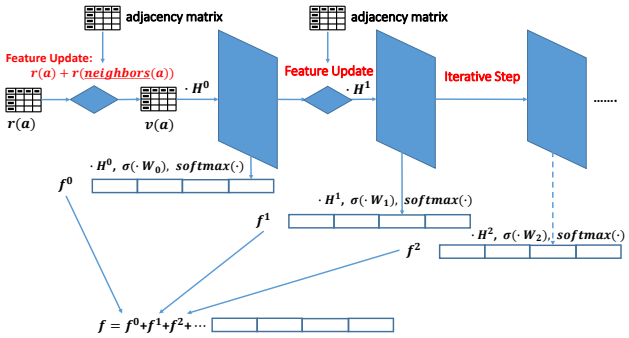


**Figure 5: Group-Role Neural Fingerprints Algorithm. "Feature Update" operation: each member's feature is updated so that the 1-hop neighbors' features and neighboring edge features are added to the initial member feature.**

learned from the training process. The $\sigma$ and *softmax* functions are given as:

$$\sigma(x) = \frac{1}{1+e^{-x}}, \quad softmax(\mathbf{z}) = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \text{ for j=1...K.}$$

Since it is a convolutional network-like structure, the number of hidden units at each hidden layer and the length of fingerprints f are both $m$. For experiments, we choose $m$ to be 10 and the size of input layer to be 6 (equals to the length of role distribution vector), but the results are robust to these choices of values.

**Algorithm 1:** Group-Role Neural Fingerprints

**Input** : group's social graph, members-role matrix, radius R, hidden layer weights $\mathbf{H^0}, ...\mathbf{H^R}$, output layer weights $W_0, ...W_R$, length of role distribution vector $t$, fingerprint length $m$.

**Initialize** : fingerprint vector $f_{1\times m} \Leftarrow 0_s$

**for** *each member a in social graph* **do**
  $r(a)_{1\times t} \Leftarrow$ a's role distribution vector
**end**
**for** $L = 0$ *to* $R$ **do**
  **for** *each member a in social graph* **do**
    $r(1)...r(N)$ = role distribution vectors of neighbors(a)
    $v_{1\times t} \Leftarrow r(a)_{1\times t} + \sum_{i=1}^{N} r(i)_{1\times t}$
    $r(a)_{1\times m} \Leftarrow \sigma(v_{1\times t} \cdot \mathbf{H^L_{t\times m}})$
    $f^L_{1\times m} \Leftarrow \text{softmax}(r(a)_{1\times m} W_L)$
    $f_{1\times m} \Leftarrow f_{1\times m} + f^L_{1\times m}$
  **end**
**end**
**Return** : real-valued vector f

## 3.4 Combining Group-level and Internal Features

As illustrated in the left half of our deep neural network in Figure 6, we have two types of features, namely, the group-level features

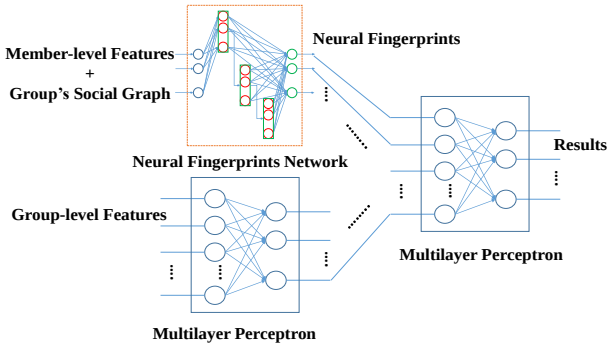


**Figure 6: Combining Group-level Features and Internal Group Features**

and our convolution of the member-level features with the group's social graph (Group-Role Neural Fingerprints). We now proceed to explore different ways to combine the group-level features and internal features to make better prediction than only using one channel of them. We investigate three combination methods:

- Method 1: Combine independent group-level and member-level predictions. Predict group popularity using group-level and member-level features independently, then combine the predictions as $\alpha\times$(group-level prediction)+(1-$\alpha$)$\times$(member-level prediction), where $\alpha \in [0, 1]$. The value of $\alpha$ is chosen over validation data.
- Method 2: Clustered adjusted combination. We first cluster the groups based on the group-level features and find the optimal $\alpha$ weight for each cluster. Whenever a newly formed group arrives, we calculate the inverse of the distances from the new group to the existing group clusters to find the optimal weight for the new sample group. The optimal $\alpha^*$ is calculated as:

$$\alpha^* = \frac{\sum_{i=1}^{N} \alpha_i / d_i}{\sum_{i=1}^{N} 1/d_i}. \quad (1)$$

  where $d_i$ is calculated as the average distance between the new group and all groups in cluster $i$.
- Method 3: Deep neural network based combination. As illustrated in Figure 6, a deep neural network is used to combine group-level and member-level features. It is a combination of our neural fingerprints network with two Multilayer Perceptrons (MLPs). It is the best method in our experiments.

# 4 PERFORMANCE EVALUATION

To evaluate the performance of our proposed method, we conduct experiments on the Meetup dataset. In section 4.1 we provide a brief description about our dataset. Then we test the prediction power of group-level features, member-level features, and group-role neural fingerprints respectively in Section 4.2. In Section 4.3, we compare the accuracy of the three methods of combining group-level features and member-level features with three competitive baselines. Finally in Section 4.4 and 4.5, we analyze the importance of various types of member roles along with the interaction patterns (social structures) between the roles for predicting group success.

## 4.1 Dataset Description

We crawled all Meetup groups located within 50 miles of New York City (NYC), from March 2003 to February 2015, including all the related meta-data. Table 2 summarizes the salient statistics of the collected dataset.

| Name | Value |
|---|---|
| Number of groups | 17,234 |
| Number of users | 1,101,336 |
| Number of events | 1,025,719 |
| Number of RSVPs | 8,338,382 |
| Number of venues | 93,643 |
| Avg. Members per group | 274.13 |
| Avg. Groups a user joins | 3.54 |
| Avg. Events per group | 72.26 |
| Avg. Participants per event | 5.67 |
| Avg. Events per active user | 9.38 |

**Table 2: Dataset Statistics**

## 4.2 Group Popularity Prediction

In this section, we show how member-level and group-level features can be used to predict the popularity of groups at different prediction intervals. Unlike the experiment settings introduced by [12], [17] and [19], which include all groups of different sizes and ages, we only focus on predicting future RSVP numbers of new groups in each year. In our experiments, features are extracted from the first three months starting from the time a newly formed group held its first event. We then make prediction of the RSVP number within another time window of three months in the future after a time interval ranging from one month to ten months. The predicted RSVP numbers are tested against the true RSVP numbers. We use the Root Mean Squared Error (RMSE) to measure RSVP prediction accuracy:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2},$$

where $y_i$ is a target group's actual RSVP number and $\hat{y_i}$ is the predicted RSVP number. Since we only focus on predicting the RSVP number of newly created groups, the RMSE is **not** dominated by single very large group in our dataset. In datasets where a few large groups dominated the RMSE, we recommend using the normalized RMSE as a metric of accuracy. After filtering out the groups without valid information to calculate the features, we have more than 7,000 new groups along with their features and RSVP numbers. We choose the first 80% groups along the time line as training set and the rest to be the testing set.

*4.2.1 Group-role Neural Fingerprints vs. Raw Member Features.* To demonstrate the Group-role Neural Fingerprints (GRNF) can truly improve the prediction accuracy and avoid the influence of group-level features, we first conduct group popularity prediction with raw member features using the classic machine learning methods (Linear Regression, Support Vector Regression, Multilayer Perceptron, and Random Forest). We then input the raw member features with adjacency matrix of the social graph to our proposed

group-role neural fingerprints algorithm. The parameters for training GRNF include: number of training epochs (50), batch size (100), learning rate (exp(-1)*3), activation function (relu), L1 regularization (exp(-4)), L2 regularization (exp(-4)), fingerprint length (10), and convolution layer sizes (10*10*10). Finally, the obtained GRNF are input to MLP to predict group popularity.

We can see from Table 3 that Group-role Neural Fingerprints can significantly reduce the prediction errors of raw member feature based predictions at different prediction intervals (1 to 10 months). The performance improvement is statistically tested by the t-test scores presented in last two columns. The performance gain of GRNF is due to its capability of taking advantage of the structural information within a group.

*4.2.2 Comparison with Baseline Methods.* Most studies on group popularity prediction, such as [12], [19], and [20], directly use group-level features or aggregate member-level features into group-level features. We compare our method which uses both group-level features and internal features (shown in Table 1) with three competitive baselines:

- Baseline 1 [12]: in addition to meta information about the groups, such as "number of group members", "number of events" and "group join mode" etc., it also uses the averaged member-level features, such as "average event attendance of members" and "standard deviation of event attendance of members" etc. "Structural features" are introduced based on a bipartite graph generated by events and users, without distinguishing the user types.
- Baseline 2 [19]: it demonstrates that structural features like triads counts and clustering coefficients have strong predictive power for predicting the longevity of the group's lifecycle in an online social messaging network.
- Baseline 3 [20]: it uses epidemic model of differential equations to fit the evolution curve of group's popularity. One advantage of this model is that it provides decent accuracy by only using the time series of daily active users (DAU). This simple baseline acts as a sanity-check to whether the time series of the group's past popularity (the number of RSVPs) alone could determine the group's future popularity.

For baseline 1&2, we implement most of the original features and apply four classical machine learning algorithms: Linear Regression, Support Vector Regression, Multilayer Perceptron, and Random Forest, and choose the one with the best performance as the representative for each baseline. For baseline 3, we use the DAU of a group in the first three months to fit the curve and the rest time to test the accuracy.

The results in Table 4 show that our final proposed approach (Combination Method 3) clearly outperforms all baselines in all prediction horizons, ranging from predicting the average 3-month RSVP numbers in the immediate next three months to predicting this quantity ten months after the last record in the training data. As expected, for all methods, the error of predicting nearer future is smaller. Thus, it is important to contrast the accuracy gains of our method against all baselines, which range from 3.91% to 12.32%. The statistical significance of the performance improvement is verified by the p-values reported in the last column of Table 4. We see similar results for different averaging windows (4 and 5 months). Due to space limit, we don't include the results here.

*4.2.3 Impact of Different Combination Methods.* Since our predictions are made through two independent feature sets, we can combine features using different methods proposed in Section 3.4. In Table 5 we compare the accuracies of the three combination methods. For method 2, we try two clustering methods: K-means and DBSCAN (choose the one with better performance). We can see that MLP adjusted combination yields the best average RMSE=101.65 for all prediction intervals, improving the best baseline (baseline 1) by 11.3%.

| Combining Method | Method 1 | Method 2 | Method 3 |
|---|---|---|---|
| Average RMSE | 108.74 | 108.49 | 101.65 |
| Gain from Best Baseline (%) | 5.41 | 5.42 | 11.3 |

**Table 5: Performance of Three Combination Methods**

## 4.3 Member Role Analysis

In this section, we analyze the results produced by role discovery method in Section 3.3.2 and try to answer the question: "*who are the members contributing the most to a group's future popularity?*"

As detailed in Section 3.3.2, we can get the role distribution vector for a group by aggregating the role distribution vectors of all its members. We then calculate the Pearson correlation between each group's RSVP number and its role distribution. In addition, every member is assigned to a role according to the largest element in her role distribution vector. By averaging the feature vectors of all members assigned to each role, we obtain a representative feature vector for each role to study user's typical behavior.

Inspired by [7], we infer and interpret each role by analyzing the representative feature vector for each role and comparing each role's "own-feature part" and its "neighbor-feature part" in its representative feature vector, that is, we infer the role of a user by analyzing "who she is" and "who she knows". For example, if we observe a user has many more connections with her friends (reflected by the degree of the node representing the user in the social graph defined in Definition 3.2) than all her friends have, then we know she is probably the most the active user in her local social network, and we can further infer that there is a good chance that she is an event/group organizer in reality. In our experiments, we set the role number to six and get the representative feature vectors for the six roles.

**Role A (Group Organizers):** The user potentially interacts with a large number of people (average own degree=3,568.24) and is more active than her neighbors (average neighbor's degree=473.14). Together with other information about Role A's behaviors, such as joining very few groups (average group number=1.12) and attending as many events as possible (attended event fraction=0.73), we infer that the user is potentially a successful event organizer. We also find that 33.6% of the users assigned to Role A have hosted (sent the first RSVP "yes") at least one event, which is higher than the percentages in other roles. The correlation between Role A and group RSVP number is 0.514.

**Role B (Inactive Followers):** Compared with other roles, the user's neighbors are much more active than the user herself (average own degree=2.19, while average neighbors' degree=5,550.03),

**Table 3: RMSE of Raw Member Features vs. RMSE of Group-role Neural Fingerprints (GRNF)**

| Interval | Raw Member Features | | | | GRNF | Gain(%) | \|t-statistic\| | p-value |
|---|---|---|---|---|---|---|---|---|
| | LR | SVR | MLP | RF | | | | |
| 0m | 93.49 | 93.81 | 177.08 | 94.08 | **89.49** | 4.27 | >1.14 | <0.009 |
| 1m | 99.19 | 98.93 | 111.45 | 101.68 | **89.95** | 9.08 | >1.17 | <0.009 |
| 2m | 105.55 | 104.88 | 154.12 | 112.31 | **97.11** | 7.41 | >1.15 | <0.01 |
| 3m | 111.02 | 111.01 | 249.73 | 116.96 | **101.17** | 8.86 | >1.07 | <0.016 |
| 4m | 118.31 | 118.49 | 195.26 | 120.48 | **104.59** | 11.59 | >1.17 | <0.014 |
| 5m | 121.51 | 121.41 | 208.91 | 120.32 | **110.05** | 8.53 | >1.26 | <0.008 |
| 6m | 129.97 | 129.86 | 132.32 | 128.11 | **115.81** | 9.59 | >1.69 | <0.005 |
| 7m | 142.41 | 141.11 | 342.78 | 144.32 | **128.36** | 9.03 | >1.33 | <0.008 |
| 8m | 147.55 | 146.85 | 139.82 | 146.52 | **132.87** | 4.97 | >1.24 | <0.01 |
| 9m | 148.47 | 148.46 | 359.89 | 151.69 | **134.56** | 9.35 | >1.11 | <0.01 |
| 10m | 153.33 | 153.66 | 172.28 | 154.63 | **142.56** | 7.02 | >1.13 | <0.009 |

| Horizon (months) | Baseline 1 | | | | Baseline 2 | | | | Baseline 3 | Proposed Method | Gain over Best Baseline (p-value) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | LR | SVR | MLP | RF | LR | SVR | MLP | RF | | | |
| 0–3mo | 93.33 | 90.38 | 83.92 | 86.71 | 99.33 | 95.45 | 94.61 | 217.48 | 103.53 | **74.78** | **10.90%** (<0.012) |
| 1–4mo | 99.71 | 97.07 | 89.85 | 94.72 | 107.11 | 101.31 | 119.67 | 162.86 | 105.44 | **80.32** | **10.61%** (<0.012) |
| 2–5mo | 107.89 | 103.71 | 99.81 | 99.97 | 113.76 | 107.45 | 131.78 | 130.96 | 107.64 | **88.62** | **10.68%** (<0.015) |
| 3–6mo | 112.84 | 109.62 | 109.68 | 106.15 | 115.38 | 113.11 | 207.69 | 176.77 | 108.71 | **95.96** | **8.53%** (<0.022) |
| 4–7mo | 118.78 | 116.91 | 111.18 | 103.91 | 120.41 | 120.63 | 108.74 | 171.81 | 108.61 | **99.84** | **3.91%** (<0.021) |
| 5–8mo | 119.93 | 120.32 | 110.51 | 117.58 | 121.68 | 124.07 | 111.13 | 175.71 | 108.32 | **96.69** | **10.73%** (<0.012) |
| 6–9mo | 123.96 | 128.99 | 114.38 | 118.01 | 126.01 | 132.93 | 117.69 | 201.73 | 117.28 | **99.85** | **12.69%** (<0.009) |
| 7–10mo | 136.37 | 141.94 | 126.52 | 131.75 | 139.67 | 145.85 | 139.55 | 240.74 | 126.41 | **110.83** | **12.32%** (<0.011) |
| 8–11mo | 142.79 | 147.49 | 133.41 | 134.11 | 150.28 | 151.58 | 172.21 | 278.53 | 138.61 | **118.07** | **11.51%** (<0.014) |
| 9–12mo | 146.24 | 149.26 | 137.75 | 139.36 | 156.95 | 153.27 | 188.18 | 346.88 | 140.54 | **122.04** | **11.39%** (<0.014) |
| 10–13mo | 151.11 | 155.09 | 143.83 | 144.22 | 161.69 | 158.51 | 151.24 | 268.63 | 153.67 | **126.86** | **11.42%** (<0.011) |

**Table 4: Prediction Accuracy (RMSE) of Group's Future RSVP Number**

indicating that role B may be an inactive user who only participates in the events with the user's friends who have strong social influence. The correlation with group RSVP number is 0.515.

**Role C (Conference Audience):** The user is normally just a participant of a seminar. Large average own degree (4206.95) and behavior similarity with her neighbors suggest that Role C may be one of the conference audience. This role has the lowest correlation with the RSVP number (correlation=0.482).

**Role D (Inactive Users):** Similar to role B, but the user's neighbors have less interactions with others and her neighbors have less "influence" than Role B. So role D may be an inactive member who follows her own interest. The correlation with group RSVP number is 0.623.

**Role E (Small Group Organizers):** The user seems like a *scaled-down* version of role A, with lower own and neighbor activity levels. This user could be an event organizer who specializes in small events. The correlation with group RSVP number is 0.555.

**Role F (Dedicated Followers):** Compared to other roles, this user has high activity level (average own degree=225.35) and potentially prefers attending social-based events than conference-based events, since the average event size (225.35/10.39=21.68) she attends is much lower than Role C (4206.95/70.58=59.61). Also, the high activity level of her neighbors (average neighbor degree=1578.18) suggests that Role F may represent the dedicated followers of the "social influencers". This role has the highest correlation with the RSVP number (correlation=0.700).

## 4.4 Social Structure Analysis

We have shown in the previous section that members' roles can affect a group's popularity. In this section we show that the interaction patterns between various roles can also contribute to a group's popularity. One important advantage of our proposed method is that we can trace from the bit in the output fingerprint which contributes the most to the RSVP number back to the exact substructure that activates the bit most, thus find the most important substructure. For this Group-role Neural Fingerprint analysis, we select the members whose related features activate the hidden units the most at each radius, then use their connections with neighbors to represent the subgraphs contributing the most to the group's success. In Figure 7 we show some sampled subgraphs at radius=1, each member is labeled with her assigned role. All the sampled subgraphs come from the most popular groups whose 3-month RSVP numbers are ranked within top 100 among all groups in Meetup. The structures of their social graphs in the first 3 month exhibit

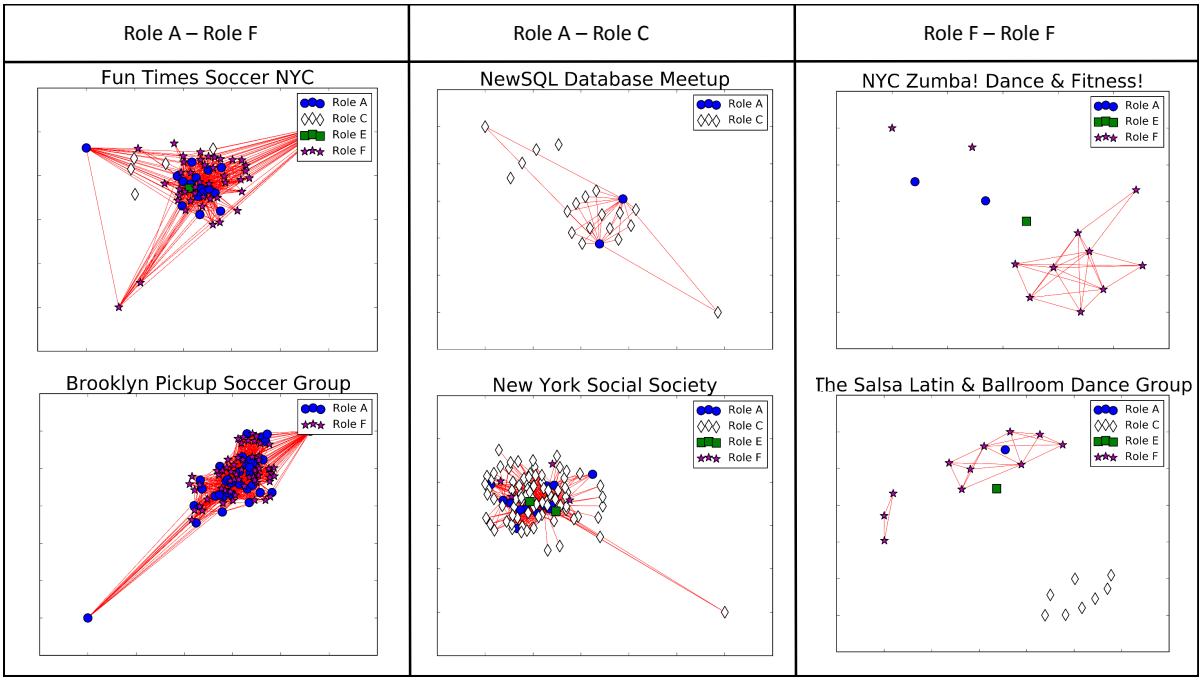


**Figure 7: Subgraphs Contributing the Most to Group Popularity. Three types of learned subgraphs (radius=1) are: role A-F, role A-C, and role F-F. For each subgraph type, two example groups with high popularity (top-100 popular in Meetup) that contain this type of subgraph are shown.**

clear "Role A - Role F", "Role A - Role C" or "Role F- Role F" pattern. On the other hand, when we examine unpopular groups, we couldn't find the previous three clear graph patterns.

**Role A (Organizers) - Role F (Followers):** Normally seen in social-based groups such as popular soccer groups. There is an organizer (role A) for each game, and several active players (role F). Role C (Conference Audience) seldom exists in such group.

**Role A (Organizers) - Role C (Audience):** Frequently seen in technical seminars. A large number of role C (Audience) members exist in the group with only one or two organizers. The intensity of social connections between members in this type of groups is weaker than what is common in more social-based groups.

**Role F (Followers) - Role F (Followers):** Normally seen in dancing groups. Similar with role A-F structure, the social intensity in this type of groups tends to be strong. They differ in that the connections among ordinary members are stronger.

Interestingly, we find that topics of the groups are highly correlated to the role distribution and structures of their social graphs. For example, soccer groups like "*Fun Times Soccer NYC*" and "*Brooklyn Pickup Soccer Group*" tend to have a lot of "Role A-Role F" structures, technical seminars like "*New SQL Database Meetup*" and "*New York Social Society*" normally are full of "Role A-Role C" structures, and dancing groups, such as "*NYC Zumba! Dance and Fitness*" and "*The Salsa Latin&Ballroom Dance Group*", are likely to have more "Role F-Role F" structures. On the other hand, we can see that if a group organizer wants to gain more group popularity in the future, she probably needs to build up such interaction patterns in an early stage.

## 5 CONCLUSION

In this paper, we proposed a deep neural network method to predict the future popularity of groups in event-based social networks. Our method outperformed all state-of-the-art methods. Along the way, we have analyzed a few key factors contributing to these predictions. Specifically, we showed that location and time are important group-level features. We also demonstrated that member roles and interaction among members with different roles, characterized by neural fingerprints, can better represent the intrinsic member behaviors and the social structure of a group than the raw member features and the activity graph. Through case studies in Meetup, we showed that the most relevant user roles to a group's popularity are not "organizer-like members", but "ordinary members". We also found that the most important substructures are not combining all the most important roles, but follow different combination patterns for different types of groups.

## REFERENCES


[1] Jie Bao, Yu Zheng, David Wilkie, and Mohamed Mokbel. 2015. Recommendations in location-based social networks: a survey. *GeoInformatica* 19, 3 (2015), 525–565.

[2] Hao Ding, Chenguang Yu, Guangyu Li, and Yong Liu. 2016. Event Participation Recommendation in Event-Based Social Networks. In *International Conference on Social Informatics*. Springer, 361–375.

[3] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 135–144.

[4] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. 2015. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in neural information processing systems*. 2224–2232.

[5] Sean Gilpin, Tina Eliassi-Rad, and Ian Davidson. 2013. Guided learning for role discovery (glrd): framework, algorithms, and applications. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 113–121.

[6] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 855–864.

[7] Keith Henderson, Brian Gallagher, Tina Eliassi-Rad, Hanghang Tong, Sugato Basu, Leman Akoglu, Danai Koutra, Christos Faloutsos, and Lei Li. 2012. Rolx: structural role extraction & mining in large graphs. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1231–1239.

[8] Houda Khrouf and Raphaël Troncy. 2013. Hybrid event recommendation using linked data and user diversity. In *Proceedings of the 7th ACM conference on Recommender systems*. ACM, 185–192.

[9] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[10] Daniel D Lee and H Sebastian Seung. 2001. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*. 556–562.

[11] Xingjie Liu, Qi He, Yuanyuan Tian, Wang-Chien Lee, John McPherson, and Jiawei Han. 2012. Event-based social networks: linking the online and offline social worlds. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1032–1040.

[12] Xiang Liu and Torsten Suel. 2016. What makes a group fail: Modeling social group behavior in event-based social networks. In *Big Data (Big Data), 2016 IEEE International Conference on*. IEEE, 951–956.

[13] Augusto Q Macedo, Leandro B Marinho, and Rodrygo LT Santos. 2015. Context-aware event recommendation in event-based social networks. In *Proceedings of the 9th ACM Conference on Recommender Systems*. ACM, 123–130.

[14] Meetup 2017. Meetup. (2017). Retrieved October 31th, 2017 from https://en.wikipedia.org/wiki/Meetup_(website)

[15] Annamalai Narayanan, Mahinthan Chandramohan, Lihui Chen, Yang Liu, and Santhoshkumar Saminathan. 2016. subgraph2vec: Learning distributed representations of rooted sub-graphs from large graphs. *arXiv preprint arXiv:1606.08928* (2016).

[16] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 701–710.

[17] Soumajit Pramanik, Midhun Gundapuneni, Sayan Pathak, and Bivas Mitra. 2016. Can i foresee the success of my meetup group?. In *Advances in Social Networks Analysis and Mining (ASONAM), 2016 IEEE/ACM International Conference on*. IEEE, 366–373.

[18] Zhi Qiao12, Peng Zhang, Chuan Zhou, Yanan Cao, Li Guo, and Yanchun Zhang. 2014. Event recommendation in event-based social networks. (2014).

[19] Jiezhong Qiu, Yixuan Li, Jie Tang, Zheng Lu, Hao Ye, Bo Chen, Qiang Yang, and John E Hopcroft. 2016. The lifecycle and cascade of wechat social messaging groups. In *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 311–320.

[20] Bruno Ribeiro. 2014. Modeling and predicting the growth and death of membership-based websites. In *Proceedings of the 23rd international conference on World Wide Web*. ACM, 653–664.

[21] Leonardo FR Ribeiro, Pedro HP Saverese, and Daniel R Figueiredo. 2017. struc2vec: Learning node representations from structural identity. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 385–394.

[22] David Rogers and Mathew Hahn. 2010. Extended-connectivity fingerprints. *Journal of chemical information and modeling* 50, 5 (2010), 742–754.

[23] Ryan A Rossi and Nesreen K Ahmed. 2015. Role discovery in networks. *IEEE Transactions on Knowledge and Data Engineering* 27, 4 (2015), 1112–1131.

[24] Yiye Ruan and Srinivasan Parthasarathy. 2014. Simultaneous detection of communities and roles from large networks. In *Proceedings of the second ACM conference on Online social networks*. ACM, 203–214.

[25] Zhenhua Wang, Ping He, Lidan Shou, Ke Chen, Sai Wu, and Gang Chen. 2015. Toward the new item problem: context-enhanced event recommendation in event-based social networks. In *European Conference on Information Retrieval*. Springer, 333–338.

[26] Hongzhi Yin, Yizhou Sun, Bin Cui, Zhiting Hu, and Ling Chen. 2013. Lcars: a location-content-aware recommender system. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 221–229.

[27] Tianyang Zhang, Peng Cui, Christos Faloutsos, Yunfei Lu, Hao Ye, Wenwu Zhu, and Shiqiang Yang. 2016. Come-and-go patterns of group evolution: A dynamic model. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1355–1364.

[28] Wei Zhang, Jianyong Wang, and Wei Feng. 2013. Combining latent factor model with location features for event-based group recommendation. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 910–918.

[29] Yao Zhang, Bijaya Adhikari, Steve TK Jan, and B Aditya Prakash. 2017. Meike: Influence-based communities in networks. In *Proceedings of the 2017 SIAM International Conference on Data Mining*. SIAM, 318–326.

[30] Vincent W Zheng, Sandro Cavallari, Hongyun Cai, Kevin Chen-Chuan Chang, and Erik Cambria. 2016. From node embedding to community embedding. *arXiv preprint arXiv:1610.09950* (2016).