

Group 45: CS224w Project Milestone Report

Yilun Wang Guoxing Li
yilunw@stanford.edu guoxing@stanford.edu

Jingrui Zhang
jingrui@stanford.edu

November 12, 2014

Abstract

Event recommended systems are usually based on the users past history of event attending and also similar users. Such traditional recommended systems suffer from three problems. The first is the cold start problem. In many cases, the system has no knowledge about the interests of a new user. Second, the data sparsity problem that some users may have a few attended event. Third, these traditional approaches fail to take network structure of a user into consideration. However, a noticeable trend in social life is that people sharing characteristics and attributes tend to stay friends so that knowledge about the friends of a person demonstrates his preference in certain degree. Thus the social network would be expected to provide useful and extra information of a user besides his similarity to other users. To address the aforementioned three problems, we proposed a Hybrid Approach Based on Network Structure and Collaborative Filtering approach. The random walk based network structure mining method discovers the underlying network structure while collaborative filtering mitigates the 'cold start' and data sparsity problem.

1 Introduction

Recommendation engine has been widely used these days, especially for online social media, as it is a key factor to deliver personalized content and improve user experience. Recommendation engine attempts to recommend item of interest based on previous behavior or basic information of each user. However, as the recommendation engine is used in social network, few recommendation methods take into consideration the underlying network structure in a social network system. We intend to explore different approaches to evaluate to what extent the network structure combined with network information can help us accurately recommend events to users.

The goal of the project is to predict what events users will be interested in based on user friendship, user actions, event metadata, and demographic

information. Our final deliverable would be a event recommendation engine. The input of the system is organized into five part: (1) User-event (1) User (2) Events (3) Event attendees and (4) user-user connections. The output of the system is a list of events ordered from the ones in which you predict the user will be most interested to those in which the user will be least be interested.

2 Data

The raw data is from an social network app based on events in Kaggle Startup Program[1]. The format of the data is described below,

- User event: Each row indicates an event that was shown to a user, which contains user, event, invited, and timestamp.
- User basic info: User metadata including user_id, locale, birth year, gender, joined time, location, and timezone.
- User friend: User friendship constructs a undirected network while the users represent the nodes and the friendship relationships between users represents the edges. We further study the the basic statistic of the network.

# Nodes	15897435
# Edges	30279252

- Events basic info: Events data contains the specific details of the event like creator of the event, start_time of the event, city, state, zip, country, lat and lng.
- Event attendee: Event attendee contains information about which users attended various events. It is the information about the certainty they are going to the event: yes, maybe, invited, and no. Yes, maybe, invited, and no are representing users who indicated that they were going, maybe going, invited to, or not going to the event.

3 Related Work

In [2], the authors present a social network-based recommendation system that uses data from user profiles and user-to-user connections. The basic idea is to utilize PMF to find a low-rank decomposition of R , where R_{ij} is the rating user i gave to game j and approximate it as a product of two low-rank matrices $R \approx UG$. These would result in the traditional recommended system based on similarity. To incorporate in the social network factors, the author utilized an extra matrix F , where $F_{i,j} = 1$ if user i listed user j as a geek buddy and zero otherwise, to account in the social network factors, $R \approx FUG$.

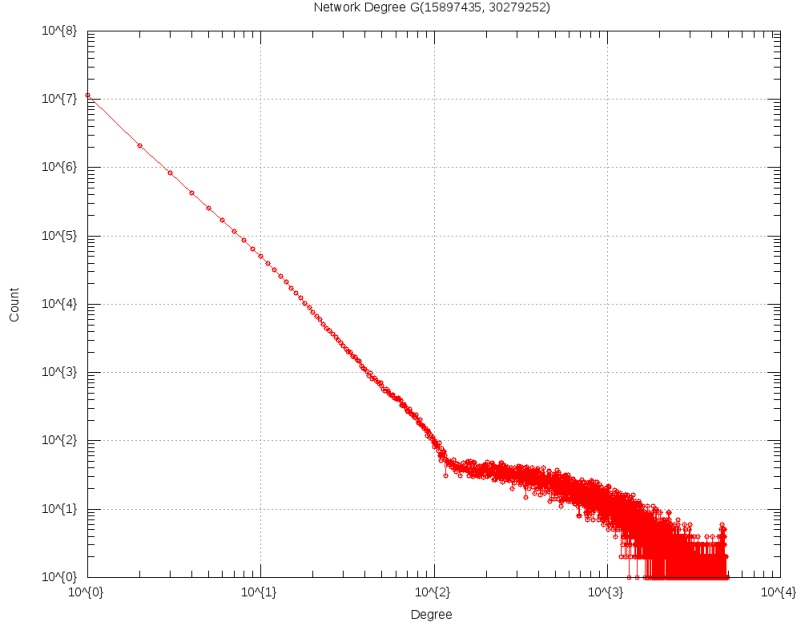


Figure 1: Degree Distribution of User Network

However, simply using 1 as a friend connection and 0 as lack of a connection would be a very rough assumption that each connection would have similar influence on that user. Intuitively, the person closer to the user would know better of the user and have more similar taste in certain aspects. The closer connection would make good and useful recommendations while loose connection would less likely to be qualified to do so. Thus closeness of connection should be account into algorithms.

[3] presents a novel supervised learning algorithm, Supervised Random Walk, to predict/recommend links in social networks. The algorithm basically works as follows. It first combines the network structure and the characteristics (attributes, features) of nodes and edges of the network together to assign a strength to each edge, which is then used to model the random walk transition probability. After calculating that, the algorithm starts a random walk at s , which will give each node u a probability p_u . Nodes are then ordered by p_u and top ranked nodes are predicted as destinations of future links of s . To find the optimal edge strength function, the authors define an optimization problem.

The paper proposed a novel idea that incorporates network structure information. However, the algorithm requires a significant amount of knowledge of existing edges to make accurate prediction while unsupervised link prediction usually doesnt need this. Also, for link recommendation, it would be great for the authors to take the activeness of nodes into consideration. For example, a

person with more friends will be more open to making new friends, and thus should be recommended more potential friends.

[4] presents a hybrid approach of event recommendation based semantic web which contains a content-based system of Linked data and a collaborative filtering system of social information.

This paper assumes that there is a sufficiently number of past attended events for every user to avoid the cold-start problem. However, for a real-world dataset, 'cold-start' is a common problem of many recommendation methods. Some users might only have a few events they attended in the past and could lead to bad performance in content-based recommendation. For collaborative filtering part, the authors consider the co-attendances of the user and its friends. This suffers a lot from the sparsity problem of the data. A matrix factorization approach can both consider the co-attendances and somehow solve the sparsity problem.

4 Algorithms and Model

We plan to design and evaluate several algorithms. The hybrid approach based on network structure and collaborative filtering is assumed to achieve the best result while other algorithms serve as baselines.

4.1 Supervised Random Walk

The detail of the algorithm has been explained in previous section. We are particularly interested in applying the algorithm to our event dataset since the link that we are trying to predict is between different types of node. We would also like to explore different approaches to incorporate time of link formation into prediction. For example, time of event creation could be treated as a feature, or newer edges will be assigned a higher strength. Supervised Random Walk exploit the underlying network structure in our dataset.

4.2 Traditional Machine Learning

It will be interesting to see how Supervised Random Walks performs compared with traditional machine learning algorithms. In this case, we model the recommendation system as a classification problem where given a pair of user and event, we predict whether the user is interested in the event. We plan to try different combination of features, and use logistic regression as the learning algorithm.

4.3 Collaborative filtering

We construct a matrix, with users as the columns and events as the rows. We can design a matrix factorization based method to predict the score of an event for a specific user based on the latent factor we learned from the matrix.

4.4 Hybrid Approach Based on Network Structure and Collaborative Filtering

As in [4], recommendation suffers from 'cold start' problem. Also, many recommendation techniques fail to take the network structure into consideration. To solve these problems, we need to fully exploit the user similarity information and the network structure information. Therefore, we proposed to combine the traditional collaborative filtering approach with random walk based network structure mining approach .

5 Evaluation and Test

For result evaluation, we will use the 30% of the our dataset for testing (while keeping 70% of the data for training). We are going to use the same evaluation criteria for our multiple approaches and make a fairly detailed comparison.

The output format would be in (user_id, event_id) format ordered by the intensity of interest of a user with user_id to an event with event_id.

The evaluation metric for this project is Mean Average Precision (MAP) . For more information, please refer to Appendix A.

6 Reference

- [1] Event Recommendation Engine Challenge. <https://www.kaggle.com/c/event-recommendation-engine-challenge>
- [2] Aranda, Jorge, et al. "An online social network-based recommendation system." Toronto, Ontario, Canada (2007).
- [3] Backstrom, Lars, and Jure Leskovec. "Supervised random walks: predicting and recommending links in social networks." Proceedings of the fourth ACM international conference on Web search and data mining. ACM, 2011.
- [4] Khrouf, Houda, and Raphal Troncy. "Hybrid event recommendation using linked data and user diversity." Proceedings of the 7th ACM conference on Recommender systems. ACM, 2013.

A Mean Average Precision

Suppose there are m missing outbound edges from a user in a social graph, and you can predict up to 10 other nodes that the user is likely to follow. Then, by adapting the definition of average precision in IR, the average precision at n for this user is:

$$ap@n = \sum_{k=1}^n P(k) / \min(m, n) \quad (1)$$

where if the denominator is zero, the result is set zero; $P(k)$ means the precision at cut-off k in the item list, i.e., the ratio of number of users followed up to the position k over the number k , and $P(k)$ equals 0 when k -th item is not followed upon recommendation; $n = 10$

(1) If the user follows recommended nodes #1 and #3 along with another node that wasn't recommend, then $ap@10 = (1/1 + 2/3)/30.56$

(2) If the user follows recommended nodes #1 and #2 along with another node that wasn't recommend, then $ap@10 = (1/1 + 2/2)/30.67$

(3) If the user follows recommended nodes #1 and #3 and has no other missing nodes, then $ap@10 = (1/1 + 2/3)/20.83$

The mean average precision for N users at position n is the average of the average precision of each user, i.e.,

$$MAP@n = \sum_{i=1}^N ap@n_i / N \quad (2)$$