!Trendz: Recommender System using Facebook Profile

Ramakrishnan Kandhan and **Nikhil Teletia**

Abstract

Online communities have been growing at an exponential rate since the advent of social networking sites such as Orkut and Facebook. Such websites house a wealth of information about a particular individual such as their likes, dislikes, location and their network of friends. In this report we discuss a new approach for a universal recommender system which uses the information available about a person in Facebook to suggest content.

Introduction

Recommender Systems, which refer to information filtering techniques that attempt to present information content that are likely of interest to the user, have been widely used in sites such as Amazon.com for many years now. Most of the research in this area has been concentrated on content (news, blogs, movies etc.) specific suggestion systems (Ricardo and Joaquim 2004), (Lai and Ku 2003), (Daniel and Sean 2005). The few efforts to build generic systems have relied on extensive user feedback and hence have not been successful (Yih and Vitor 2006), (Raymond and Terence 2007). Here we suggest the use of the information available in Facebook instead of user feedback to build a universal recommender system.

In Facebook, users voice out their interests/opinions using several channels such as fan pages, groups, wall posts etc. However Facebook has a stringent privacy policy which doesn't allow access to personal information without authorization (Atif and Chen-Nee 2008). Hence, we focus on "groups", which have unrestricted access.

A group is a collection of users who have a common trait/interest. Each user can be a part of one or more groups which represent a subset of his/her interests. A group also has some information regarding its content such as the type, subtype that it falls under and a brief description about its goals/use. We crawled through Facebook and extracted around 22,000 groups which we used as our dataset.

Approach

Our goal for this project was to setup a framework for developing suggesting systems, which make use of an individuals Facebook profile. We make no assumptions about the content that is being suggested, So this system can be used to suggest content within Facebook such as groups, fan pages apps, ads etc. or generic web content such as blogs, videos, music etc.

Recommender systems, typically define some characteristics for a user based on the information available, and seeks to predict the rating(probability) that a user would like an item they had not yet considered. These characteristics could be a ranked list of the user's likes/dislikes or interests or the user's social environment (the collaborative filtering approach). Successful recommender systems generally perform a combination of the two to improve accuracy. Here we suggest a method to perform Collaborative filtering as well as to obtain a ranked list of interests for a user based on his/her Facebook profile.

Ranked List of Interests

The goal here could be stated as follows, given a set of interests assign a rating (probability) for each of these interests for a user based on the information available. We use the subtype field of groups in Facebook as our vocabulary of interests. There are 141 unique subtype fields in our dataset. The simplest way to obtain the ranked list is to pool together the groups of a particular user and then rank the interests based on the number of groups under each subtype that the user is part of. However, this approach has some glaring disadvantages. Suppose the user is only part of groups of 20 subtypes, all other subtypes are given a score zero and hence there is no way in which we can distinguish between them even though the user might have a clear preference for one over the other. Even when a user is part of an equal number of groups for two subtypes, this approach doesn't give us any way of finding out which subtype the user prefers more. The trouble with this approach is the assumption that the users profile is complete which is not the case. The group list of a user represents only a partial list and hence we must first complete this list before performing the ranking. In order to complete the group list of a user we compute the Group Weight Matrix, which encodes information about the correlation between two groups.

Let $g_1, g_2, ..., g_N$ and $u_1, u_2, ..., u_M$ represent the N groups and M users in the dataset. Let $U_1, U_2, ..., U_N$ represent the user list for each group and $G_1, G_2, ..., G_M$ represent the group list for each user. Also let $n_1, n_2, ..., n_N$ represent the number of users in each group and $m_1, m_2, ..., m_M$ represent the number of groups for each user. Given this setup, we first infer statistically significant associations between pairs of groups. Intuitively since group g_i has n_i users and Group g_j has n_j we expect the number of common users for g_i, g_j to be $E(i,j) = (n_i * n_j)/N$. If the actual number of common users between these two groups $(A(i,j))$ deviates significantly from this expected value, the assumption that the two groups are independent is questionable and we say that the two groups are correlated. This effect can be easily captured by the χ^2 test.

$$\chi^2 = \frac{(E(i,j)-A(i,j))^2}{E(i,j)} + \frac{(E(\bar{i},j)-A(\bar{i},j))^2}{E(\bar{i},j)} + \frac{(E(i,\bar{j})-A(i,\bar{j}))^2}{E(i,\bar{j})} + \frac{(E(\bar{i},\bar{j})-A(\bar{i},\bar{j}))^2}{E(\bar{i},\bar{j})}$$

In this equation $A(\bar{i},j)$ represents the actual number of users not present in g_i but present in g_j and so on. Now we can use this test to find if two groups are correlated, however it cannot judge its strength. Hence when we find that two groups are correlated we calculate its strength(weight) as follows.

$$w_{ij} = \sum_{u_k \epsilon U_i \cap U_j} (\frac{1}{m_k}) \quad (1)$$

The group weight matrix W, calculated as given above is a $N*N$ sparse matrix. We let R_0 to be the group rank list for a particular user u_i for whom we are interested in calculating the ranked list. R_0 is a vector of length N where each element k is set to 1 if $g_k \epsilon U_i$ and 0 otherwise. Now we calculate the new rank list from R_0 using W using the formula given below. We stop the iteration when $R_t - R_{t-1} < \epsilon$. Note that though this approach looks quite similar to the power iteration approach in PageRank we cannot use the same approximation here since the assumption that W has a unique leading eigenvector and that R_0 have a nonzero component in the direction of the leading eigenvector is not valid in this case.

$$R_{t+1} = W * R_t \quad (2)$$

The final group rank list that we compute here contains the rating for each group for user u_i. From this we compute the rating for each subtype by the summation of the rating of all groups belonging to that particular subtype. Figure 1 shows the tag cloud representation for the interests of a particular user.

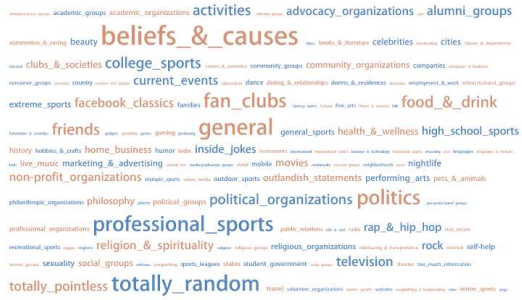

Figure 1: Tag Cloud for subtypes as interests

Each subtype also has a type associated with it, which can be used to compute the rating for more general interests as shown in figure 2.

Collaborative Filtering

Collaborative filtering is based on the principle that those who agreed in the past tend to agree in the future. So the task here is to find a set of k users with similar tastes and then base the rating for content based on the rating given by these k 'allies'. Collaborative filtering is widely used in sites such as Netflix, Amazon where the prior purchases/ratings are used to find 'allies' for a particular user. In a social networking context the friends of a particular user are the obvious choice when it comes to choosing 'allies'. However there may be other users not in the friends list who might be a better 'ally', since friendship is not based only on similar interests. Also not all friends are equally helpful when it comes to picking content for a user. So hence we need some metric to measure the similarity between two users to perform Collaborative filtering.

Figure 2: Tag Cloud for types as interests

The simplest metric could be the final group rank list calculated in eqn 2. We could find the top k allies by using the k-nearest neighbour algorithm (Shakhnarovish and Indyk 2005). However this could be very cumbersome as the dimensions of this vector are huge. A good approximation would be to use the rating for each interest instead to find the 'allies'. This measure though intuitive is not perfect. For example, two users who like movies could have a widely different taste within movies which is not captured in this metric.

To overcome this we cluster groups based on the weight graph, computed in eqn 1 (and not based on subtype as done before) and then calculate the rating for each cluster by summing up the rating for each group in the cluster from the final group rank list calculated in eqn 2. The grouping is performed by using spectral clustering. From the cluster rating vector, we use k-nn to find the k closest allies for a particular user. Intuitively, the clustering of groups based on the weight graph puts groups with a high number of common users together and hence one would expect the topics of these groups to be correlated. Thus, these clusters are expected to be a better approximation. These k 'allies' of a user can then be used to rate the content that a user will like by using any of the standard algorithms such as Slope One (Daniel and Anna 2005).

Conclusion and Future Work

In this report, we presented an approach to use the information available about a person in Facebook to generate a ranked list of interests and also to find users with similar taste, which could then be used to develop recommender systems for a wide variety of content. In future we plan on extending this approach to rate keywords rather than interests which could be very helpful in rating web content such as blogs, news etc.

References

Atif, N; Saqib, R., and Chen-Nee, C. 2008. Unveiling facebook: A measurement of study of social network based applications. *JMC*.

Daniel, L., and Anna, M. 2005. Slope one predictors for online rating-based collaborative filtering. *SDM*.

Daniel, L., and Sean, M. 2005. Implementing a rating based item-to-item recommender system. *ICEC*.

Lai, H.J; Liang, T., and Ku, Y. 2003. Customized internet news services based on customer profiles. *ICEC*.

Raymond, KP; Alfonso, F. D. J., and Terence, J. 2007. iscore: Measuring the interestingness of articles in a limited user environment. *CIDM*.

Ricardo, C.; Jaime, M. D. G., and Joaquim, A. 2004. Evaluating adaptive user profiles for news classification. *IUI*.

Shakhnarovish, M; Darrell, K., and Indyk, K. 2005. Nearest-neighbor methods in learning and vision. *The MIT Press*.

Yih, W; Joshua, G., and Vitor, R. 2006. Finding advertising keywords on web pages. *WWW*.