

Project Report

TEAM MEMBERS

Guangmo Tong (net-ID: gxt140030)
Ruili Yao (net-ID: rxy121130)
Xinwen Zhu (net-ID: xxz126730)

SELECTED TOPIC

In this project, we apply big data techniques and machine learning algorithms on Twitter dataset using Spark.

PROJECT DESCRIPTION

Today, more and more people are looking for friends through the online social network. Therefore, the k-friend recommendation problem has become quite attractive in this field for the related companies such as Twitter. That is, how to recommend k users sharing similar interest to a certain user on Twitter.

The purpose of this project is to implement two approaches of friends recommendation on Twitter data using Spark. The basic one uses Jaccard similarity as the main measure for the recommendation. The second one is a new approach, as we put forward in this project, called clustering-enhanced recommendation, which covers several Machine Learning techniques such as K-means algorithms, Cosine/Pearson similarity measures, etc.

RELATED FILES

There are totally seven files for this project including three datasets and four script files.

- 1. twitter_5000_10000.dat:** The 10000 who-follow-who relationships of the selected 5000 nodes in matrix format, representing as 1 or 0.
- 2. twitter_5000_10000new.dat:** The i-th line contains the nodes followed by node i, representing as node_id.
- 3. cluster.dat:** Line i contains the nodes in cluster i.
- 4. basic_jaccard.scala:** The recommendation via Jaccard similarity.

5. cluster.scale: Cluster the nodes into 100 clusters by the Euclidean distance.

6. advanced_pearson: Recalculate the attribute vector according to the clustering and do the recommendation by Pearson similarity.

7. advanced_cos: Recalculate the attribute vector according to the clustering and do the recommendation by Cosine similarity.

TECHNICAL APPROACH

1. Basic Recommendation

The friends recommendation problem is supposed to be that, given a user i , the users recommended to i should be similar to i . Here, we develop the measure of similarity based on the who-follows-who relationship from the dataset. That is, for a user i , we define its attribute vector A_i as

$$A_u = (A_u^1, \dots, A_u^N) \in \{0, 1\}^N$$

where $A_i^j = 1$ (resp. $A_i^j = 0$) implies i follows (resp. does not follow) j . We also define $A(i, i)$ to be 1.

According to the Jaccard similarity measure, the similarity $S(i, j)$ of two users i and j is

$$S(i, j) = \frac{\sum_{l=1}^N \min(A_i^l, A_j^l)}{\sum_{k=1}^N \max(A_i^k, A_j^k)}. \quad (1)$$

Thus, a natural approach is to recommend to j the k users that have the largest $S(i, j)$, which results in the following output.

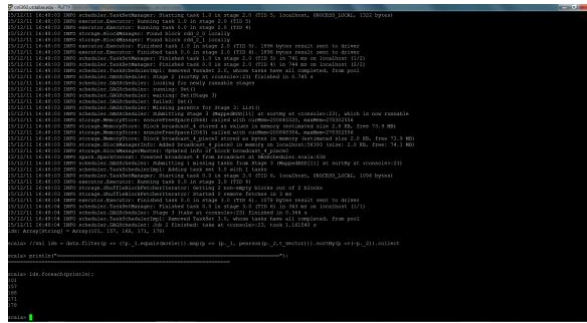


Figure 1. Result of basic_jaccard.scala

However, as we all know, a node can be referred to several topics on Twitter, i.e., sports, news and music. In this case, the basic approach may magnify a certain topic too much. For example, suppose there are ten users u_1, \dots, u_{10} with the following who-follows-who relationship shown in figure 2, let's consider select one user from u_2 and u_3 to recommend to u_1 .

Based on Eq.(1), it's easy to conclude that $S(u_1, u_2) = 5/7 > S(u_1, u_3) = 4/7$, which means u_2 is a better choice. But, if we have known that u_4, u_5, u_6 and u_7 are all music pages, and u_8, u_9 and u_{10} are related to different other topics, then we'll make quite a different decision since u_1 has two common interest with u_2 while four with u_3 .

$$\begin{matrix} & u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 & u_8 & u_9 & u_{10} \\ \begin{matrix} u_1 \\ u_2 \\ u_3 \end{matrix} & \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \end{matrix}$$

Figure 2. Example of who-follow-who relationship

Therefore, to address this problem, we propose another approach, namely clustering-enhanced recommendation.

2. Clustering-enhanced Recommendation

2.1 Topic Clustering

First, clustering the users into 100 clusters by the k-means algorithm, where the distance of two users is calculated by the Euclidean distance.

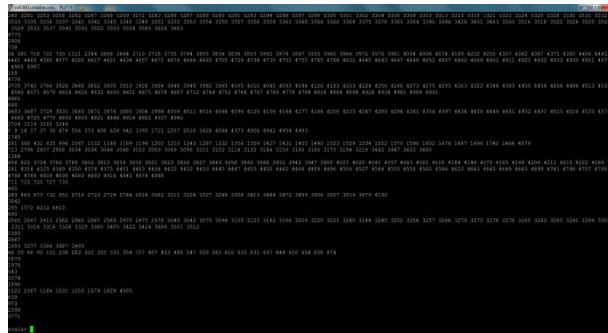


Figure 3. Result of cluster.scala

2.2 Friend Recommendation

Then, we can redefine the attribute vector of each user based on the clusters shown in figure 3 as follows.

$$\bar{A}_i = (\bar{A}_i^1, \dots, \bar{A}_i^p),$$

where \bar{A}_i^j is the number(1/0) of users in cluster C_j followed by user i .

Finally, we can compute the similarity of two users by the Pearson correlation coefficient or Cosine similarity measure according to this newly defined attribute vector.

(Please refer to the demo PPT for the Pearson correlation/Cosine similarity formula.)

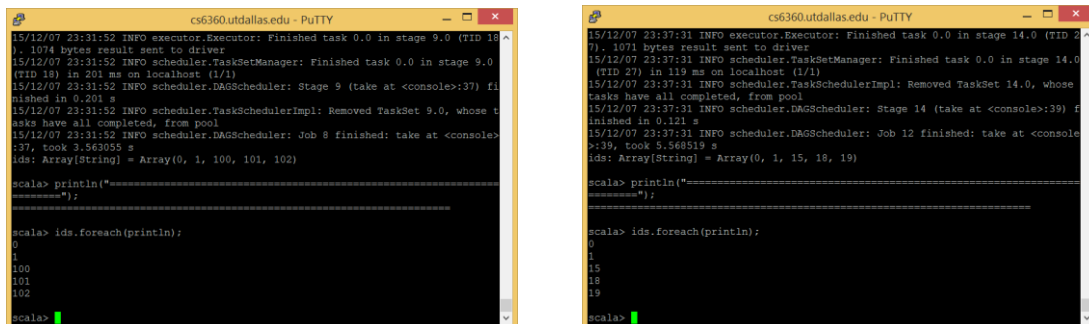


Figure 4. Results of advanced_pearson.scala / advanced_cos.scala (user_id=10)