ISSN: 2664-9640

Research on friend recommendation algorithm based on FP-Growth

Sijin Lu

School of China University of Mining & Technology-Beijing, Beijing 100000, China.

1454582323@qq.com

Abstract

Aiming at the problem of recommending the friends and various information that users need in the college student social applications, the FP-Growth algorithm in data mining is used to achieve accurate friend recommendation, and FP-Tree is generated by mining the relationship between users. And then mine frequent itemsets based on FP-Tree, and finally determine the support of selecting friends, so as to recommend the closest friends to the user. The main research direction of this paper is to describe the algorithm in detail and give examples to prove its application.

Keywords

Social applications, data structure, FP-Growth, FP-Tree.

1. Research background

With the continuous development and progress of the Internet, the social platform, an important product of the Internet era, has become an important part of people's lives. It has not only changed the way people obtain information, but also changed the way of communication between people. People can socialize Activities such as entertainment, information acquisition, and publishing of one's own views on a certain thing are carried out on the platform. College students are in the transitional stage of study and work. They rely more on social platforms to obtain information and communicate with others. The current strong social software such as QQ and WeChat are aimed at people of all classes and all ages[1], so we have to design a social applications only for college students, and the core of this applications is its recommendation algorithm. Through the recommendation algorithm, we can more accurately recommend people related to him for each student and find him useful information. This article will focus on the FP-Growth algorithm to implement our designed social applications friend information recommendation system.

2. FP-Growth algorithm implementation

2.1. Algorithm overview

Apriori and FP-Growth algorithms are now mainly used by various social software to mine association rules between users, recommend users associated with them, and help them obtain useful information. The Apriori algorithm is the earliest used association rule mining algorithm. The algorithm is divided into two steps. One is to find all frequent itemsets of users; the other is to build association rules between users [2], so the algorithm needs to scan the database multiple times. Many frequent items are generated, which greatly increases the burden of memory operation. The FP-Growth algorithm mainly uses the concept of tree in the data structure to mine data sets, compresses the database that provides frequent itemsets to a frequent pattern tree, and obtains frequent patterns by mining the frequent pattern tree [3].

ISSN: 2664-9640

The main task of the FP-Growth algorithm for mining user association rules is to find frequent items in the data set, which is mainly divided into the following three steps:

(1)Scan the database storing user information, determine the associated user information, generate a frequent item set, and then obtain the calculation method of the support degree through the algorithm, calculate the support degree of each frequent item, and then calculate the support degree of each frequent item according to the support degree. The items are sorted in descending order, and finally a list of items that meet the conditions is generated.

(2)According to the degree of support, the users selected in the first step are sorted according to the degree of support, and a new item list is generated. Insert each item in the new item table one by one into the FP-Tree rooted at null, and finally generate the FP-Tree we need.

(3)Mining frequent itemsets based on the FP-Tree generated in the second step to obtain the association rules we need.

The pseudo code of the FP-Growth algorithm is as follows:

```
void FP_growth(Tree tree){
if Tree contains a single path point then{
for all combinations of nodes in the path point group
Generate mode group and tree;
Support = The minimum support of nodes in the group;
} else {
for each tree i at the head of the tree (support from low to high) {
Generate mode treei and tree;
Support = treei.support;
Establish the conditional mode base of mode and construct the conditional mode tree TreeR;
if TreeR is not empty then
Call FP_growth (TreeR mode);
}
}[4]
```

2.2. Algorithm application

(1) Set the minimum support of each item in the collection to 2.

(2)The first operation that needs to be performed is to determine the selected frequent itemsets. The frequent itemsets selected in this article are shown in the Items column in Table 1, and then the support of each frequent item in the selected frequent itemsets is obtained by calculation. All items below the set support are deleted, and finally the itemsets are sorted in descending order to get the set { b:5, c:5, e:4, a:4, d:2 }, the results are shown in Table 1.

Table 1: Friendship data set

TID	Items	Ordered Items
001	b,d,c,f,g,i	b,c,d
002	b,c,e,h	b,c,e
003	b,c,e,z	b,c,e
004	a,b,o,p	b,a
005	b,w,h,m	b,a,d
006	a,e,c	c,e,a
007	a,c,e,y,q	c,e,a

(3) The most critical step is to use the steps in the above algorithm description to construct the FP-tree we need, and insert each frequent item of Ordered Items into an FP-Tree whose root node is null. If a frequent item node already exists in the FP-Tree during the insertion operation of a frequent item, the support of the node in the tree corresponding to the frequent item needs to be increased by 1; If the node in the tree corresponding to the frequent item does not exist, it is necessary to create a node with a support degree of 1 corresponding to the frequent item set, and link the node to the item table.

(4)After constructing FP-Tree, it is necessary to perform frequent pattern mining operations on FP-Tree. The mining operation needs to use the FP_growth algorithm we mentioned above. If the selected item set is a single-path pattern tree, the permutation and combination method in mathematics can be used to obtain the frequent pattern of the suffix tree contained in the tree. Here is an example, the mode suffix a is selected, and the conditional mode base of a can be determined by observing the tree as (b:2) and (c e:2). The two conditional mode bases can be used to combine to obtain the set { b:2, c:2, e:2 }, the support of all items in the set satisfies the condition that the support is greater than or equal to 2, so you can directly use the method of permutation and combination in mathematics to get the frequent pattern with the pattern suffix e as { b a:2,c a:2,e a:2,c e a:2 }.

(5) After solving the problem of the single-path conditional pattern tree, it is also necessary to solve the problem of the multi-path conditional pattern tree. For example, by observing FP-Tree, it can be determined that the conditional pattern tree of e is a multi-path pattern tree, and the conditional pattern bases of e are (b c:2) and (c:2). For the multi-path conditional pattern tree, we first need to use the method of solving the single-path conditional pattern tree problem, and obtain a set of frequent patterns of e { c e:4, b e:4, b c e:2 }through permutation and combination. Then the function FP_growth is called recursively, knowing that the pattern suffix is empty-end, the result of the recursive call is that the pattern suffix is { c,e }, the conditional pattern base { b:2 } of the pattern suffix is a single-path conditional pattern tree, and then the combination is obtained { b c e:2}. Finally, the frequent pattern of e is { c e:4, b e:2, b c e:2 }.

(6) The frequent pattern table of friends whose minimum generation support is not less than 2 is shown in Table 2.

Table 2: Friends frequent pattern table

Item	Conditional pattern base	Frequent pattern
d	{ (b a:1), (b c:1) }	{ b d:2 }
e	{ (b c:2), (c:2) }	{ c e:4, b e:4, b c e:2 }
a	{ (b:2), (c e:2) }	{ b a:2, c a:2, e a:2, c e a:2 }
c	{ (b:3) }	{ b c:3 }

From the friend frequent pattern table, we can get the association rules between different users. Their frequent pattern and support represent their association relationship. For example, { b d:2 } means that the user has both b and d friends at the same time, and the support is 2. { c e:4 } means that the user has both c and e friends at the same time, and the support is 4. We can mine all the algorithms that meet the support conditions based on the FP-growth algorithm, and then recommend the users with the most connected friends to sort out the complex relationships in social networks and recommend friends who may be helpful to the user and information.

3. Conclusion

This article mainly introduces the FP-growth algorithm and gives an application example, builds a friend recommendation model based on association rules, and generates a friend

recommendation model table, which can better recommend friends for users. The FP-growth association algorithm used in this article is based on the user's friend relationship for data mining and has generated frequent patterns. In actual application scenarios, it is more credible and accurate, and can recommend users who are closest to them.To help users get more useful materials and information, and to enhance user experience and experience. However, the FP-growth algorithm used in this article does not correlate and mine other samples of user friends, so in future research and development, more samples of friends should be drawn to recommend friends for target users more accurately.

Acknowledgements

China University of Mining & Technology-Beijing innovation training program.

References

[1] Y.G. Li, S.H. Yao, G.X. Wang, W. Feng: Investigation and analysis of social media use and opinion expression of College Students (Popular, Standardization, (2021) No. 3), p.138. (In Chinese)

[2] C.Q. Xiong,X. Chen: Research on social friend recommendation method based on FP growth (Journal Of Hubei University Of Technology, Vol. 35 (2020) No. 1), p.34-35. (In Chinese)

[3] C.G Han: Research and improvement of FP growth algorithm (Journal of Longyan University, Vol. 38 (2020) No. 5), p.8. (In Chinese)

[4] Information on https://blog.csdn.net/Seek_Joy/article/details/52601404.