Massive datamining, Excercise 2, Huu V.Le

March 22, 2020

1 About

1.1 Notebook

Massive data mining, TDTU Spring 2020
****Excercise 2: Friend recommendation****

### Warning:
- This notebook was built localy by jupyter notebook not any online notebook like colab or kaggle lab.
- It's mean that, if you run localy you must be installed all needed packages bellow.
- I not sure that the codes bellow will completely run on online notebook tools.

1.2 Author

- Name: **Huu V.Le**
- ID: **51703095**

2 Preprocessing

2.1 Import packages

[3]:
```
from pyspark.sql import *
from pyspark.sql.types import *
from pyspark.sql.functions import *
from pyspark import SparkContext

from operator import add

import pandas as pd
```

2.2 Create spark session and context

[4]:
```
# create the Spark Session
spark = SparkSession.builder.getOrCreate()

# create the Spark Context
sc = spark.sparkContext
```

```
[ ]:
```

2.3 Load data.

- [USER][TAB][FRIEND1, FRIEND2,..]

Let's define schema for that data. Two columns: user_id and friends. - user_id: id of target user - friends: list friend of target user, seperated by ',' formated as string

```
[5]: schema = StructType([
         StructField("user", StringType()),
         StructField("friends", StringType()),
     ])
```

Let's read data with defined schema

```
[6]: data = spark.read\
                 .option('delimiter', '\t')\
                 .csv('./data/soc-LiveJournal1Adj.txt', schema=schema)
```

```
[7]: data.show(10)
```

```
+----+--------------------+
|user|             friends|
+----+--------------------+
|   0|1,2,3,4,5,6,7,8,9…|
|   1|0,5,20,135,2409,8…|
|   2|0,117,135,1220,27…|
|   3|0,12,41,55,1532,1…|
|   4|0,8,14,15,18,27,7…|
|   5|0,1,20,2022,22939…|
|   6|0,21,98,2203,3238…|
|   7|0,31993,40218,404…|
|   8|0,4,38,46,72,85,2…|
|   9|  0,6085,18972,19269|
+----+--------------------+
only showing top 10 rows
```

```
[ ]:
```

3 Alogrithm, solution

3.1 Problem

!!!Vietnamese!!!

Thuật toán: Sử dụng thuật toán đơn giản sau: Với mỗi user U thuật toán sẽ kiến nghị N=10 người không là bạn của U nhưng có số lượng bạn chung với U nhiều nhất.

3.2 Instructions:

1. Get list friends of friends of input user from dataframe. **Note that this is list as string not python list**.
2. We need to flat map the above string to list friend id.
3. Remove the user id in flatmap that is input user friend.
4. The problem return to wordcount => count user id in the flatmap.
5. Sort by value descending

3.2.1 Functions

[144]:
```python
def friend_recommender(user_id):
    list_friend = data.filter("user = '{}'".format(user_id)).
↪collect()[0]['friends'].split(',')
    result = data.select('friends')\
        .filter(col('user').isin(list_friend))\
        .rdd.flatMap(lambda x: x).flatMap(lambda x: x.split(','))\
        .filter(lambda x: not x in list_friend)\
        .filter(lambda x: x != str(user_id))\
        .map(lambda x: (int(x), 1))\
        .reduceByKey(add)\
        .map(lambda x: (x[1], x[0]))\
        .takeOrdered(10, lambda x: (-x[0], x[1]))
    return [x[1] for x in result]
```

3.3 Tesing:

Để kiểm tra thuật toán của bạn đúng bạn có thể so sánh kết quả của bạn với danh sách gợi ý cho user ID 11 là (top 10): 27552, 7785, 27573, 27574, 27589, 27590, 27600, 27617, 27620, 27667.

[145]:
```python
friend_recommender('11')
```

[145]:
```
[27552, 7785, 27573, 27574, 27589, 27590, 27600, 27617, 27620, 27667]
```

- Year, It's right!

3.4 Make result

[146]:
```python
users = [924, 8941, 8942, 9019,9020, 9021, 9022, 9990, 9992, 9993]
```

[147]:
```python
list_result = dict()
```

```
[148]: %%time
       for user in users:
           list_result[user] = friend_recommender(str(user))
```

```
CPU times: user 214 ms, sys: 57 ms, total: 271 ms
Wall time: 4.16 s
```

```
[149]: list_result
```

```
[149]: {924: [439, 2409, 6995, 11860, 15416, 43748, 45881],
        8941: [8943, 8944, 8940],
        8942: [8939, 8940, 8943, 8944],
        9019: [9022, 317, 9023],
        9020: [9021, 9016, 9017, 9022, 317, 9023],
        9021: [9020, 9016, 9017, 9022, 317, 9023],
        9022: [9019, 9020, 9021, 317, 9016, 9017, 9023],
        9990: [13134, 13478, 13877, 34299, 34485, 34642, 37941],
        9992: [9987, 9989, 35667, 9991],
        9993: [9991, 13134, 13478, 13877, 34299, 34485, 34642, 37941]}
```

Wow, it's light fast, isn't it?

3.4.1 Let's write result to file

```
[152]: f = open("result.txt", "a")
       for user in list_result.keys():
           f.write('{}\t{}\n'.format(user, ','.join([str(x) for x in␣
        ↪list_result[user]])))
       print('Writed result successfully!')
       f.close()
```

```
Writed result successfully!
```

4 Author contact:

- If you want more details or error feedback, please contact me.
  Github: https://github.com/leehuwuj
  Facebook: https://fb.com/leehuwuj

```
[ ]:
```