

0 Global Code

In [106]:

```
%load_ext sql
import pymysql
pymysql.install_as_MySQLdb()
%sql mysql://stu1900011102:stu1900011102@162.105.146.37:43306
```

The sql extension is already loaded. To reload it, use:
%reload_ext sql

In [107]:

```
%sql use stu1900011102;

* mysql://stu1900011102:***@162.105.146.37:43306
0 rows affected.
```

Out[107]:

[]

1 Internship 5——题目15: 图的聚类系数的近似计算

1.1 创建无向图的节点表

参考给的无向图代码

In [108]:

```
%%sql
set foreign_key_checks=0;
drop table if exists Cities;
create table Cities
(
    cityid CHAR(3) NOT NULL PRIMARY KEY,
    city VARCHAR(30) NOT NULL,
    region VARCHAR(30) NULL,
    country VARCHAR(30) NOT NULL
);
set foreign_key_checks=1;

* mysql://stu1900011102:***@162.105.146.37:43306
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
```

Out[108]:

[]

In [109]:

```
%%sql
truncate table Cities; # 每次添加之前先清空table
insert into Cities(cityid, city, region, country) values
('ATL', 'Atlanta', 'GA', 'USA'),
('ORD', 'Chicago', 'IL', 'USA'),
('DEN', 'Denver', 'CO', 'USA'),
('IAH', 'Houston', 'TX', 'USA'),
('MCI', 'Kansas City', 'KS', 'USA'),
('LAX', 'Los Angeles', 'CA', 'USA'),
('MIA', 'Miami', 'FL', 'USA'),
('MSP', 'Minneapolis', 'MN', 'USA'),
('JFK', 'New York', 'NY', 'USA'),
('SEA', 'Seattle', 'WA', 'USA'),
('SFO', 'San Francisco', 'CA', 'USA'),
('ANC', 'Anchorage', 'AK', 'USA'),
('FAI', 'Fairbanks', 'AK', 'USA');
```

```
* mysql://stu1900011102:***@162.105.146.37:43306
0 rows affected.
13 rows affected.
```

Out[109]:

[]

1.2 创建无向图的边表

参考给的无向图代码

In [110]:

```
%%sql
set foreign_key_checks=0;
drop table if exists Roads;
create table Roads
(
    city1          CHAR(3) NOT NULL REFERENCES Cities,
    city2          CHAR(3) NOT NULL REFERENCES Cities,
    distance INT     NOT NULL,
    PRIMARY KEY(city1, city2),
    CHECK(city1 < city2),
    CHECK(distance > 0)
);
set foreign_key_checks=1;
```

```
* mysql://stu1900011102:***@162.105.146.37:43306
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
```

Out[110]:

[]

In [111]:

```
%%sql
truncate table Roads; # 每次添加之前先清空table
insert into Roads(city1, city2, distance) VALUES
('ANC', 'FAI', 359),
('ATL', 'ORD', 715),
('ATL', 'IAH', 800),
('ATL', 'MCI', 805),
('ATL', 'MIA', 665),
('ATL', 'JFK', 865),
('DEN', 'IAH', 1120),
('DEN', 'MCI', 600),
('DEN', 'LAX', 1025),
('DEN', 'MSP', 915),
('DEN', 'SEA', 1335),
('DEN', 'SFO', 1270),
('IAH', 'MCI', 795),
('IAH', 'LAX', 1550),
('IAH', 'MIA', 1190),
('JFK', 'ORD', 795),
('LAX', 'SFO', 385),
('MCI', 'ORD', 525),
('MCI', 'MSP', 440),
('MSP', 'ORD', 410),
('MSP', 'SEA', 2015),
('SEA', 'SFO', 815);
```

```
* mysql://stu1900011102:***@162.105.146.37:43306
0 rows affected.
22 rows affected.
```

Out[111]:

[]

聚类系数(clustering coefficient): 衡量你的两个朋友之间也互为朋友的概率

$$C = \frac{3 \times \text{网络中三角形数目}}{\text{网络中连通的三元组数目}}$$

通过邻接矩阵自乘得到三角形个数

1.3 一步可达邻接表

In [112]:

```
%%sql
set foreign_key_checks=0;
drop table if exists Neighbor1;
create table Neighbor1
(
    city1          CHAR(3) NOT NULL REFERENCES Cities,
    city2          CHAR(3) NOT NULL REFERENCES Cities,
    PRIMARY KEY (city1, city2)
);
set foreign_key_checks=1;

* mysql://stu1900011102:***@162.105.146.37:43306
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
```

Out[112]:

[]

In [113]:

```
%%sql
truncate table Neighbor1;
insert into Neighbor1(city1, city2)
    select city1, city2 from Roads;
replace into Neighbor1(city1, city2)
    select city2, city1 from Roads;

* mysql://stu1900011102:***@162.105.146.37:43306
0 rows affected.
22 rows affected.
22 rows affected.
```

Out[113]:

[]

1.4 两步可达邻接表

矩阵自乘即可

但我感觉这里用Roads生成更简单一点

In [114]:

```
%%sql
set foreign_key_checks=0;
drop table if exists Neighbor2;
create table Neighbor2
(
    city1          CHAR(3) NOT NULL REFERENCES Cities,
    city2          CHAR(3) NOT NULL REFERENCES Cities,
    PRIMARY KEY (city1, city2)
);
set foreign_key_checks=1;
```

```
* mysql://stu1900011102:***@162.105.146.37:43306
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
```

Out[114]:

[]

In [143]:

```

%%sql
truncate table Neighbor2;
# 使用replace into避免表中出现重复数据
insert into Neighbor2(city1, city2)
    select distinct A.city1, B.city2
        from Roads as A, Roads as B
        where A.city2 = B.city1 and A.city1 != B.city2;
replace into Neighbor2(city1, city2)
    select distinct B.city2, A.city1
        from Roads as A, Roads as B
        where A.city2 = B.city1 and A.city1 != B.city2;
replace into Neighbor2(city1, city2)
    select distinct A.city2, B.city1
        from Roads as A, Roads as B
        where A.city1 = B.city2 and A.city2 != B.city1;
replace into Neighbor2(city1, city2)
    select distinct B.city1, A.city2
        from Roads as A, Roads as B
        where A.city1 = B.city2 and A.city2 != B.city1;
replace into Neighbor2(city1, city2)
    select distinct A.city1, B.city1
        from Roads as A, Roads as B
        where A.city2 = B.city2 and A.city1 != B.city1;
replace into Neighbor2(city1, city2)
    select distinct B.city1, A.city1
        from Roads as A, Roads as B
        where A.city2 = B.city2 and A.city1 != B.city1;
replace into Neighbor2(city1, city2)
    select distinct A.city2, B.city2
        from Roads as A, Roads as B
        where A.city1 = B.city1 and A.city2 != B.city2;
replace into Neighbor2(city1, city2)
    select distinct B.city2, A.city2
        from Roads as A, Roads as B
        where A.city1 = B.city1 and A.city2 != B.city2;

```

```

* mysql://stu1900011102:***@162.105.146.37:43306
0 rows affected.
18 rows affected.
18 rows affected.
18 rows affected.
18 rows affected.
28 rows affected.
28 rows affected.
54 rows affected.
54 rows affected.

```

Out[143]:

[]

In [144]:

```
%sql select * from Neighbor2
```

1.5 计算三角形个数

用Neighbor1 & Neighbor2即可

由于重合边，计算结果 / 6

In [145]:

```
%%sql
drop procedure if exists countTri;
create procedure countTri(out cnt int)
begin
    select ceiling(count(*)/6) into cnt
    from (select Neighbor1.city1 as city1, Neighbor2.city2 as city2
          from Neighbor1, Neighbor2
          where Neighbor1.city1 = Neighbor2.city1 and Neighbor1.city2 = Neighbor2.city2
          ) as tmp;
end;

* mysql://stu1900011102:***@162.105.146.37:43306
0 rows affected.
0 rows affected.
```

Out[145]:

```
[]
```

In [147]:

```
%%sql
set @cntTri = 0;
call countTri(@cntTri);
select @cntTri;

* mysql://stu1900011102:***@162.105.146.37:43306
0 rows affected.
1 rows affected.
1 rows affected.
```

Out[147]:

```
@cntTri
7
```

1.6 计算三元组个数

注意重复就可以

In [118]:

```
%%sql
set foreign_key_checks=0;
drop table if exists Trigroup;
create table Trigroup
(
    city1      CHAR(3) NOT NULL REFERENCES Cities,
    city2      CHAR(3) NOT NULL REFERENCES Cities,
    city3      CHAR(3) NOT NULL REFERENCES Cities,
    PRIMARY KEY (city1, city2, city3)
);
set foreign_key_checks=1;

* mysql://stu1900011102:***@162.105.146.37:43306
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
```

Out[118]:

```
[]
```


In [134]:

```
%%sql
truncate table Trigroup;
# 使用replace into避免表中出现重复数据
insert into Trigroup(city1, city2, city3)
    SELECT distinct A.city2, A.city1, B.city2 from Roads as A, Roads as B
    where A.city1 = B.city1 and A.city2 != B.city2;
replace into Trigroup(city1, city2, city3)
    SELECT distinct A.city2, A.city1, B.city1 from Roads as A, Roads as B
    where A.city1 = B.city2 and A.city2 != B.city1;
replace into Trigroup(city1, city2, city3)
    SELECT distinct A.city1, A.city2, B.city2 from Roads as A, Roads as B
    where A.city2 = B.city1 and A.city1 != B.city2;
replace into Trigroup(city1, city2, city3)
    SELECT distinct A.city1, A.city2, B.city1 from Roads as A, Roads as B
    where A.city2 = B.city2 and A.city1 != B.city1;
```

```
* mysql://stu1900011102:***@162.105.146.37:43306
0 rows affected.
60 rows affected.
21 rows affected.
21 rows affected.
34 rows affected.
```

Out[134]:

[]

In [135]:

```
%%sql select * from Trigroup
```

```
* mysql://stu1900011102:***@162.105.146.37:43306
136 rows affected.
```

In [139]:

```
%%sql
drop procedure if exists countGroup;
create procedure countGroup(out cnt int)
begin

    select ceiling(count(*) / 2) into cnt
        from Trigroup;

end;
```

```
* mysql://stu1900011102:***@162.105.146.37:43306
0 rows affected.
0 rows affected.
```

Out[139]:

[]

In [140]:

```
%%sql
set @cntGroup = 0;
call countGroup(@cntGroup);
select @cntGroup;
```

```
* mysql://stu1900011102:***@162.105.146.37:43306
0 rows affected.
1 rows affected.
1 rows affected.
```

Out[140]:

```
@cntGroup
        68
```

1.7 计算聚类系数

$$C = \frac{3 \times \text{网络中三角形数目}}{\text{网络中连通的三元组数目}}$$

In [148]:

```
%%sql
set @res = 3 * @cntTri / @cntGroup;
select @res;
```

```
* mysql://stu1900011102:***@162.105.146.37:43306
0 rows affected.
1 rows affected.
```

Out[148]:

```
@res
0.308823529
```

In []: