Application of coordinate mapping in data visualization

Name:高楚枫 StudentID: 5140109114

Catalog

Abstract

Force guidance algorithm is the only layout algorithm used in the data visualization of ***ACEMAP***. This algorithm has many advantages and disadvantages. In this paper, I invented two new layout algorithms: ***Linear coordinate mapping algorithm*** and ***'Electron orbit model' algorithm*** to better the data visualization of the ACEMAP. The field ***PSYCHOLOGY*** is used as an example in this paper.

Key words: ACEMAP, data visualization, layout algorithm

1. Background

On the first day when I took wireless communication course, professor Wang showed us a research work conducted in his Laboratory——the ACEMAP. I was deeply fanscinated by this outstanding work.

When we are writing papers, we have to search many different papers related to our work for references. In academic map of ACEMAP, each paper is showed as a point on the map with its designed position, size and color. These points are clustered by keywords of the paper.

Here is a picture the whole academic map(***figure-1)***:

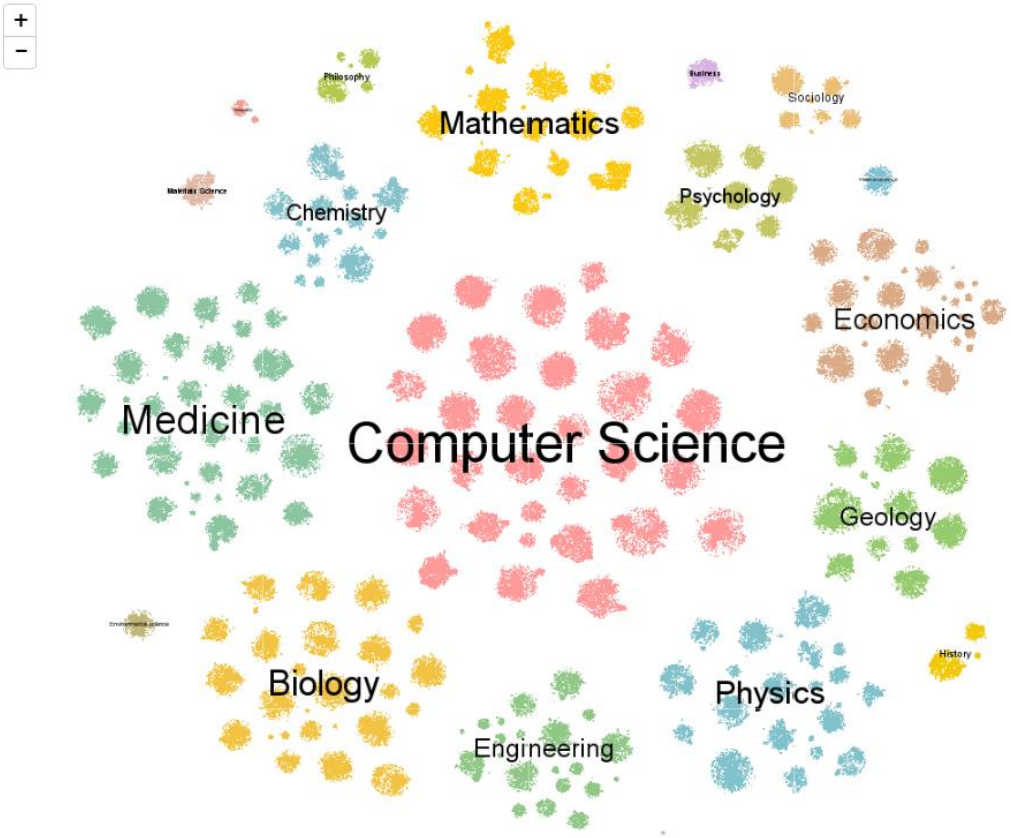

figure-1

It is really beautiful and amazing, yet there is still room for improvement. Let us call the field on map such as ***COMPUTER SCIENCE, MATHEMATICS, BIOLOGY,PSCHOLOGY, etc L0***(showed in the picture below as ***figure-2***) and call those sub-field, take the sub-fields in ***PSYCHOLOGY*** for example(showed in the picture below as ***figure-3***), such as ***PSYCHIATRY, PSYCHOTHERAPIST,NEUROSCIENCE, etc L1***. We can see that in the academic map, each L0 are independent as well as L1 which result in no connections among different clusters and plenty of blank in the map. Furthermore this academic map can not show the external influence on each paper. Those problems I have listed above are due to the present layout algorithm used.

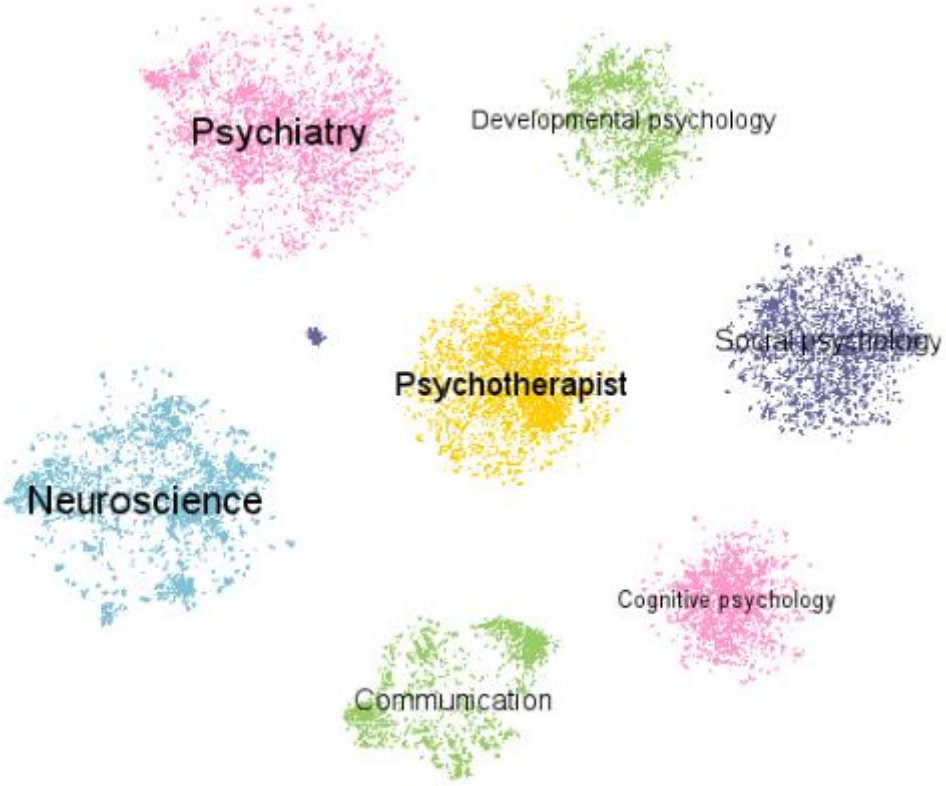

figure-2

figure-3

Therefore, I decide to invent new layout algorithms to establish the connection between different field and beautify the present academic map.

2. General idea

2.1 Use the papers in field PSYCHOLOGY as samples.

2.2 Give two methods to select points in samples.

2.3 Design two layout algorithms to map the coordinates of those selected points.

2.4 Visualize the processed points.

3. Tools, programming languages and sources of data

3.1 Tools: Pycharm, Mysql Front

3.2 Programming languages: Python2.7, SQL

3.3 Source of data: the data base of ***ACEMAP***. To be specific, I use the information of the papers in field ***PSYCHOLOGY***. All the layout algorithm I invented below are applied to the coordinates of the paper points which have been calculated by force guidance algorithm before.

4. Original layout:

In this paper we only concentrate on the position of each paper point. After leaving out the size, shape and color information, the original layout of paper points in field PSYCHOLOGY is showed in the picture below(***figure-4***):

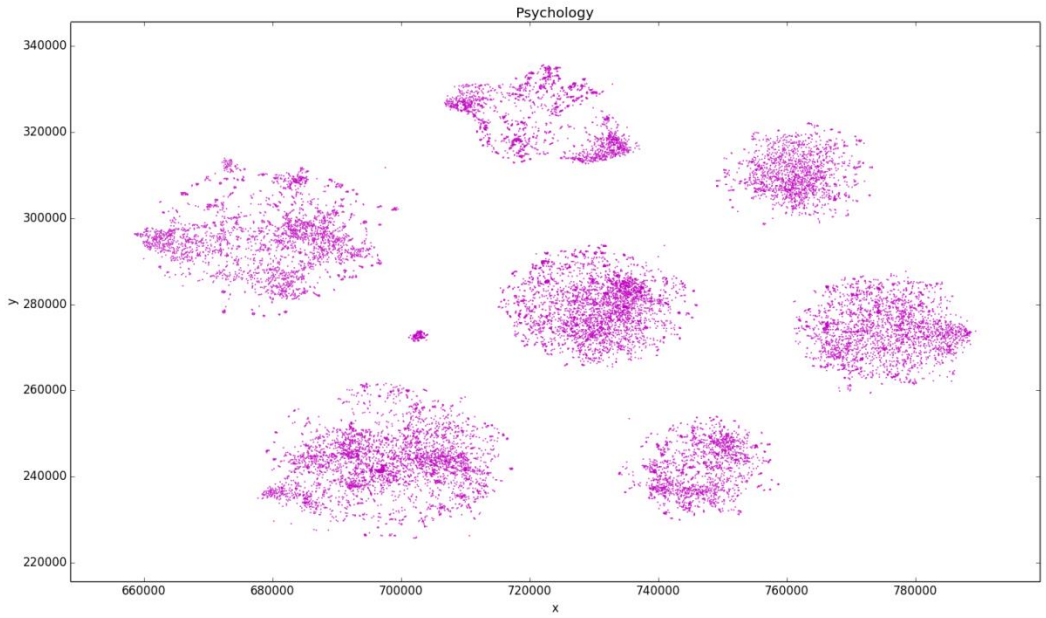

figure-4

5. Linear coordinate mapping algorithm

5.1 Process of data:

I use MYSQL FRONT to write SQL to obtain those necessary data from ACEMAP database `map-data-new` table PaperPos201704. The SQL code are showed below:

*SELECT * FROM `map-data-new`.PaperPos201704 where Field0='Psychology'*

With the code above I get the information of the ID, coordinate, Field1(L1), Field0(L0) of each paper in field PSYCHOLOGY. Part of the data are showed in the picture below(***figure-5***):

	A	B	C	D	E	F	G	H	I
1	PaperID	X	Y	R	FieldID	OmMap	Field0	Field1	
2	772D52D5	690567.6	245141.1	50.13202	0710F80A	1	Psycholog	Psychiatry	
3	7BBA7522	697124	242660.7	47.938	0710F80A	1	Psycholog	Psychiatry	
4	7A2A060F	697878.7	239074.4	50.13202	0710F80A	1	Psycholog	Psychiatry	
5	79E94010	694273.1	260065.4	51.45143	0710F80A	1	Psycholog	Psychiatry	
6	7BA29082	690650.2	244297.8	50.25334	0710F80A	1	Psycholog	Psychiatry	
7	80110D5E	691398.2	250188.7	47.938	0710F80A	1	Psycholog	Psychiatry	
8	7522F955	703115.9	236699	49.95138	0710F80A	1	Psycholog	Psychiatry	
9	7BDD6535	713886.7	249221.9	47.938	0710F80A	1	Psycholog	Psychiatry	
10	7619D68A	688658.7	247558.8	53.61487	0710F80A	1	Psycholog	Psychiatry	
11	75AAC25E	689553.3	246494.2	47.938	0710F80A	1	Psycholog	Psychiatry	
12	7F76E71B	705479.4	238651.6	50.25334	0710F80A	1	Psycholog	Psychiatry	
13	7A59521A	697258.1	241478.7	47.938	0710F80A	1	Psycholog	Psychiatry	
14	8113855B	688958.9	243719.8	59.19624	0710F80A	1	Psycholog	Psychiatry	
15	7DBAFFA5	698860.6	239231.9	47.938	0710F80A	1	Psycholog	Psychiatry	

figure-5

5.2 Points selection:

A. Find the boundary points of each L1 and Calculate the xmax, xmin, ymax, ymin:

B. Calculate the center coordinates OX, OY of each L1 with the formula below:

$OX = (xmax + xmin)/2 \quad OY = (ymax + ymin)/2$

C. Calculate the radius of each L1 with the formula below:

$$R = \max\{\left(\frac{xmax - xmin}{2}\right), \left(\frac{ymax - ymin}{2}\right)\}$$

D. Select the points the distance of which to the field center is between [R*Ration, R] with the formula below:$x^2 + y^2 \geq (R * Ration)$^2

E. If *Ration* =0.7, the selected points are showed in **red** in the picture below(***figure-6***):

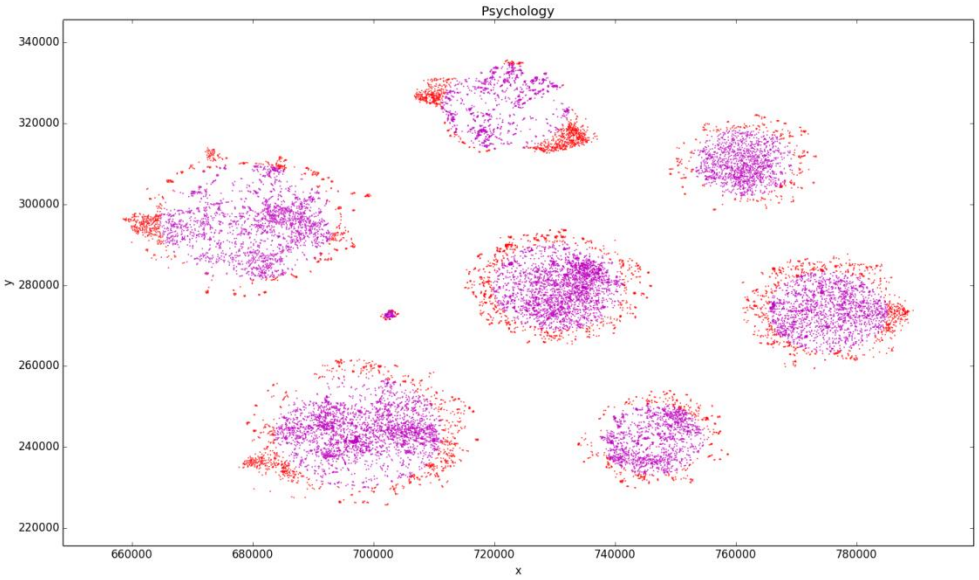

figure-6

5.3 Linear coordinate mapping algorithm

Linear coordinate map the selected points from [R*Ration, R] to [R*Ration, R*n] with the formula below:

$$d = \sqrt{(x - OX)^2 + (y - OY)\text{^}2}$$

$$\sin\theta = (y - OY)/d$$

$$\cos\theta = (x - OX)/d$$

$$Newd = (d - R * Ratio) * (n - Ratio)/(1 - Ratio) + R * Ratio$$

$$Newx = Newd * \cos\theta + OX,\ Newy = Newd * \sin\theta + OY$$

When n=2, Ratio= 0.7, the result of mapping is showed in **red** in the picture below(figure-7):

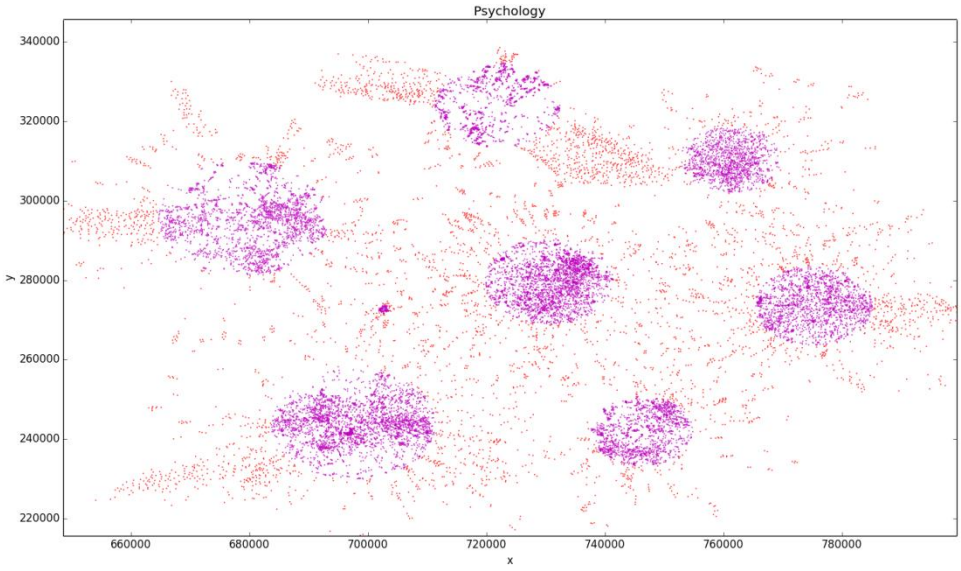

figure-7

5.4 Analysis:

Linear coordinate mapping algorithm easily disperses the boundary points and fill the blanks, making the map more beautiful.

5.5 Advantages and disadvantages:

The advantages: Linear coordinate mapping algorithm requires much smaller calculation quantity which means higher speed and the visual effect is much better.

The disadvantages: Linear coordinate mapping algorithm is in lack of practical significance and points on the boundary do not mean points of loose tie with the field1 where they belong to.

6. ‘Electron orbit model’ algorithm:

6.1 Process of data

I use MYSQL FRONT to write SQL to obtain those necessary data from ACEMAP database `map-data-new` table PaperPos201704 and table PaperReferences2. The SQL code are showed in the picture below(***figure-8***):

```
Use `map-data-new`;
#CREATE Table PSY0
#(
#SELECT A.PaperID,A.X,A.Y,A.Field1 as PaperField1,A.Field0 as PaperField0,B.PaperCitationID
#FROM
#((SELECT * FROM PaperPos201704 WHERE Field0='Psychology') as A)
#LEFT JOIN
#((SELECT PaperID as PaperCitationID,PaperReferenceID as PaperID FROM `map-data-new`.PaperReferences2) as B)
#ON A.PaperID=B.PaperID
#)


#CREATE Table PSY1
#(
#SELECT PaperID as PaperCitationID,Field1 as PaperCitationField1,Field0 as PaperCitationField0
#FROM PaperPos201704 where Field0='Psychology'
#)

SELECT PSY0.PaperID,PSY0.X,PSY0.Y,PSY0.PaperCitationID,PSY0.PaperField1,PSY0.PaperField0,
PSY1.PaperCitationField1,PSY1.PaperCitationField0
FROM PSY0
LEFT JOIN PSY1
ON PSY0.PaperCitationID=PSY1.PaperCitationID
limit 1000
```

figure-8

Part of the data I obtain are showed in the picture below(***figure-9***):

	A	B	C	D	E	F	G	H	I
1	PaperID	X	Y	PaperFiel	PaperFiel	PaperCita	PaperCita	PaperCitationField0	
2	772D52D5	690567.6	245141.1	Psychiatr	Psycholog	7E0DD922	Psychiatr	Psychology	
3	772D52D5	690567.6	245141.1	Psychiatr	Psycholog	7E0DD922	Psychothe	Psychology	
4	7A2A060F	697878.7	239074.4	Psychiatr	Psycholog	800BD3B1	Psychiatr	Psychology	
5	79E94010	694273.1	260065.4	Psychiatr	Psycholog	7A7208A6	Psychiatr	Psychology	
6	79E94010	694273.1	260065.4	Psychiatr	Psycholog	7CBACC90	Psychiatr	Psychology	
7	7BA29082	690650.2	244297.8	Psychiatr	Psycholog	755843D8	Psychiatr	Psychology	
8	7522F955	703115.9	236699	Psychiatr	Psycholog	7B6ED9B3	Psychiatr	Psychology	
9	7619D68A	688658.7	247558.8	Psychiatr	Psycholog	75492BF9	Psychothe	Psychology	
10	7619D68A	688658.7	247558.8	Psychiatr	Psycholog	75AAC25E	Psychiatr	Psychology	
11	7619D68A	688658.7	247558.8	Psychiatr	Psycholog	76498B96	Psychiatr	Psychology	
12	7619D68A	688658.7	247558.8	Psychiatr	Psycholog	7690B6C8	Psychothe	Psychology	
13	7619D68A	688658.7	247558.8	Psychiatr	Psycholog	78AF496A	Psychoana	Psychology	
14	7619D68A	688658.7	247558.8	Psychiatr	Psycholog	7A025C4D	Psychothe	Psychology	
15	7619D68A	688658.7	247558.8	Psychiatr	Psycholog	7D3ECD15	Developme	Psychology	
16	7619D68A	688658.7	247558.8	Psychiatr	Psycholog	7D61CF2D	Psychiatr	Psychology	
17	7619D68A	688658.7	247558.8	Psychiatr	Psycholog	7D61CF2D	Psychothe	Psychology	
18	7619D68A	688658.7	247558.8	Psychiatr	Psycholog	7ED41C87	Psychothe	Psychology	
19	7619D68A	688658.7	247558.8	Psychiatr	Psycholog	7EDC67F7	Developme	Psychology	
20	7619D68A	688658.7	247558.8	Psychiatr	Psycholog	7FEA1550	Psychothe	Psychology	
21	7619D68A	688658.7	247558.8	Psychiatr	Psycholog	8092E0E8	Developme	Psychology	
22	7619D68A	688658.7	247558.8	Psychiatr	Psycholog	809BA82D	Psychoana	Psychology	
23	7619D68A	688658.7	247558.8	Psychiatr	Psycholog	809BA82D	Psychothe	Psychology	

figure--9

With Python, I calculate the citation counts IC and OC of each paper both by field1 where the paper is belong and field1 where the paper is not belong.

6.2 Points selection

A. I define the 'energy' of each paper point by the formula below:

$$Energy = OC/IC$$

If IC=0, appoint Energy = 10000.

B. For each field1, sort the paper points by their energy from high to low.

C. Select the former *Ration%* of the sorted paper points as points to be

mapped.

The selected paper points are showed in **red** in the picture below(***figure-10***):

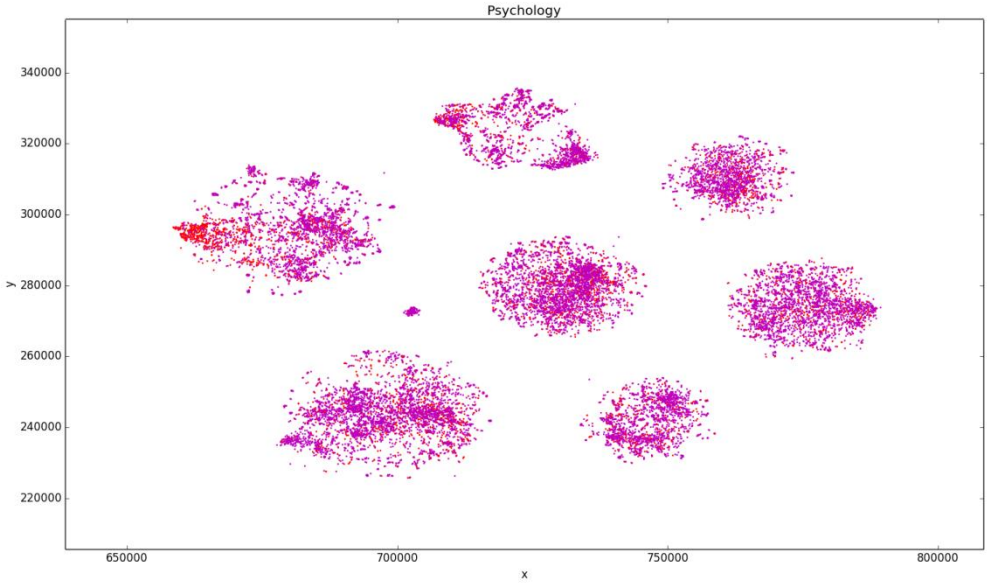

figure-10

This result indicates that the relationship between each paper point and its field1 does not lie much in its coordinates.

6.3 'Electron orbit model' algorithm

I name this coordinate mapping algorithm 'Electron orbit model' algorithm due to the similarity of this algorithm compared to the position of each electron in an atom. Electron with different energy level randomly show up in the orbit with different energy level. Those selected papers will show up in the distance region [R-R*n1,R*n2] on different energy orbit according to their 'Energy'. The high energy a paper has, the higher possibility that the paper point show in longer distance to its field1 centre is. The 'Electron orbit model' algorithm code is showed in the appendix.

When n1=0.2, n2=2, the mapping result is showed in **red** in the picture below (figure-11):

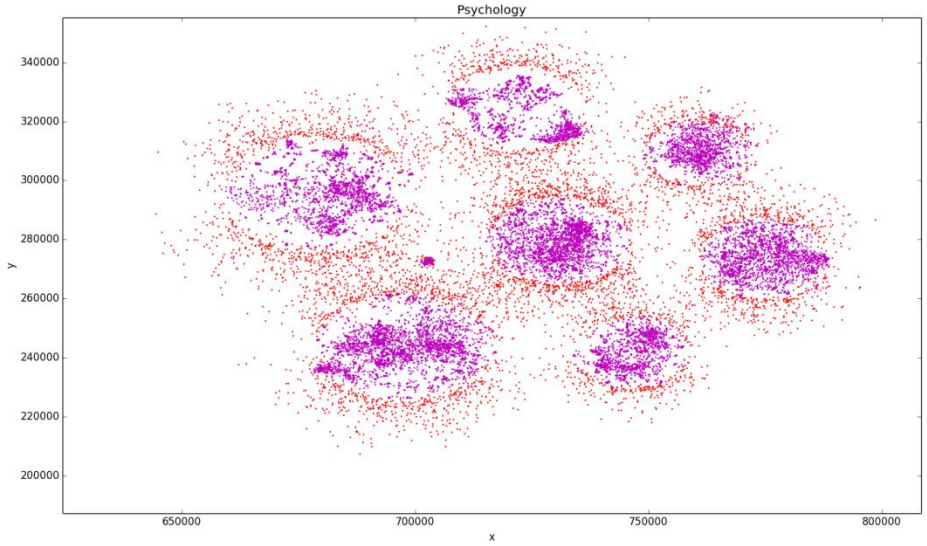

Figure-11

6.4 Analysis:

The results are as expected. Those selected paper points are dispersed beautifully. They showed in different orbit around each field1. A paper's distance to its field1 center indicates its relationship with its field1. The longer the distance is, the less relative the paper is to its field1.

6.5 Advantages and disadvantages:

The advantages: The calculation quantity of 'Electron orbit model' algorithm is very small which leads to its high efficiency. And this algorithm indicates the relationship of each paper both with its own field1 and other field1 in the same field0. Furthermore, the mapping result is beautiful.

The disadvantages: The dispersion of the selected paper points lacks in direction.

7. Comparison with force guidance algorithm and Conclusion.

Force guidance algorithm forces paper points to interact directly with one and another which makes the dispersion of paper points with strong sense of direction. However, the calculation quantity is very huge and it is impossible to process all the paper points with this algorithm. If only process part of the points with force guidance algorithm, it is meaningless.

Compared to the force guidance algorithm, ‘Electron orbit model’ algorithm concentrate on the ‘energy’ of each paper points instead of the interaction among different papers, which leads to small calculation quantity and shorten the time of calculation largely, but it only shows indirect interaction among paper points from different field1.

In conclusion, if we can combine the coordinate mapping algorithm with the force guidance calculation algorithm together, we will definitely achieve a better result.

8. Problems to be solved:

In ‘Electron orbit model’ algorithm, the process of obtaining data from the database is time-consuming, which usually takes several hours.

9. References

fig-1: http://acemap.sjtu.edu.cn/app/AcademicMap/

fig-2: http://acemap.sjtu.edu.cn/app/AcademicMap/

fig-3: http://acemap.sjtu.edu.cn/app/AcademicMap/

10. Acknowledgements

Firstly, I must thank Professor Wang Xinbing for giving us the opportunity and challenge to accomplish our individual project. During this half of semester, I have learnt quite a lot, such as how to use SQLto achieve data from the database as well as using Python to process data and visualize data. What I have learnt through this course is worth that of all the other courses add up together in a year!

Secondly, I must thank my classmates Wu Hao and Li Jiefeng. They really help me during this course. Without their guidance, I even don’t know how to start my work!

Last but not least, I must thank the three group leaders Tang Weijie, Sun Wei and Jia Yuting for their support .

11. Appendix:

11.1 Linear coordinate mapping algorithm:

```
import csv
import math
import matplotlib.pyplot as plt
import random
import heapq

def Scatter(X,Y,OX,OY,R,Ratio,n):
    d=math.sqrt((X-OX)**2+(Y-OY)**2)
    cosfun=(X-OX)/d
    sinfun=(Y-OY)/d
    d=(d-R*Ratio)*(n-Ratio)/(1-Ratio)+R*Ratio
    X=d*cosfun+OX
    Y=d*sinfun+OY
    return X,Y

def Select_Points(Xlist,Ylist):

OX=(float(heapq.nlargest(1,Xlist)[0])+float(heapq.nsmallest(1,Xlist)[0]))/2

OY=(float(heapq.nlargest(1,Ylist)[0])+float(heapq.nsmallest(1,Ylist)[0]))/2

R=(max(float(heapq.nlargest(1,Xlist)[0])-float(heapq.nsmallest(1,Xlist)[0]),flo
at(heapq.nlargest(1,Ylist)[0])-float(heapq.nsmallest(1,Ylist)[0])))/2
    SelectX1=[]
    SelectY1=[]
    SelectX2=[]
    SelectY2=[]
    for i in range(0,len(Xlist)-1):
        Ratio=0.7
```

```
                n=2
                if
math.sqrt((float(Xlist[i])-OX)**2+(float(Ylist[i])-OY)**2) >= R*Ratio:

x,y=Scatter(float(Xlist[i]),float(Ylist[i]),OX,OY,R,Ratio,n)
                    Xlist[i]=str(x)
                    Ylist[i]=str(y)
                    SelectX1.append(Xlist[i])
                    SelectY1.append(Ylist[i])
                else:
                    SelectX2.append(Xlist[i])
                    SelectY2.append(Ylist[i])
            return SelectX1,SelectY1,SelectX2,SelectY2

        def Classify_Points(A):
            Xlist1=[]
            Xlist2=[]
            Ylist1=[]
            Ylist2=[]
            flag=A[2][7]
            Xlist0=[]
            Ylist0=[]
            for i in A:
                if cmp(i[7],'Field1')==0:
                    continue
                else:
                    if cmp(i[7],flag)==0:
                        Xlist0.append(i[1])
                        Ylist0.append(i[2])
                    else:
```

```
SelectX1,SelectY1,SelectX2,SelectY2=Select_Points(Xlist0,Ylist0)
                        Xlist1.extend(SelectX1)
                        Xlist2.extend(SelectX2)
                        Ylist1.extend(SelectY1)
                        Ylist2.extend(SelectY2)
                        flag=i[7]
                        Xlist0=[]
                        Ylist0=[]
                        Xlist0.append(i[1])
                        Ylist0.append(i[2])
            SelectX1,SelectY1,SelectX2,SelectY2=Select_Points(Xlist0,Ylist0)
            Xlist1.extend(SelectX1)
            Xlist2.extend(SelectX2)
            Ylist1.extend(SelectY1)
            Ylist2.extend(SelectY2)
            return Xlist1,Ylist1,Xlist2,Ylist2

        # Psychiatry
        # Social psychology
        # Developmental psychology
        # Psychoanalysis
        # Neuroscience
        # Communication
        # Psychotherapist
        # Cognitive psychology

        def main():
            csvfile_1 = file('Psychology.csv', 'rb')
            reader_PSY = csv.reader(csvfile_1)
```

```
# csvfile_2 = file('PaperCitationCount.csv', 'rb')
# reader_PCC = csv.reader(csvfile_2)
PSY=[]
# PCC=[]
Xlist=[]
Ylist=[]

for i in reader_PSY:
    Xlist.append(i[1])
    Ylist.append(i[2])
    PSY.append(i)
del Xlist[0]
del Ylist[0]

xmax=heapq.nlargest(1,Xlist)[0]
xmin=heapq.nsmallest(1,Xlist)[0]
ymax=heapq.nlargest(1,Ylist)[0]
ymin=heapq.nsmallest(1,Ylist)[0]
print xmax,xmin,ymax,ymin
plt.title("Psychology")
plt.xlim(float(xmin)-10000,float(xmax)+10000)
plt.ylim(float(ymin)-10000,float(ymax)+10000)
plt.xlabel("x")
plt.ylabel("y")
Xlist1,Ylist1,Xlist2,Ylist2=Classify_Points(PSY)
print len(Xlist1),len(Xlist2),len(Ylist1),len(Ylist2)
plt.scatter(Xlist1,Ylist1,s=0.2,color='r')
plt.scatter(Xlist2,Ylist2,s=0.2,color='m')
# plt.scatter(Xlist,Ylist,s=0.1,color='r')
```

```
            # plt.scatter(Xlist,Ylist,s=0.2,color='m')
            plt.show()
            # plt.savefig('C:\Pycharm\example\.idea\pic.png')

            csvfile_1.close()
            # csvfile_2.close()



            # import matplotlib.pyplot as plt
            # import random
            # plt.title("I'm a scatter diagram.")
            # plt.xlim(xmax=7,xmin=0)
            # plt.ylim(ymax=7,ymin=0)
            # plt.xlabel("x")
            # plt.ylabel("y")
            # Alist=[]
            # for a in range(100):
            #         Alist.append(10*random.random())
            # Blist=[]
            # for a in range(100):
            #         Blist.append(10*random.random())
            #
            # plt.plot(Alist,Blist,'ko')
            # plt.show()

    main()
```

11.2 'Electron orbit model' algorithm:

```
    #PaperID X Y PF1 PF0 PCID PCF1 PCF0
    import csv
    import math
```

```
import matplotlib.pyplot as plt
import random
import heapq
import operator

def Select_Points(COUNTLIST):
    IC=0
    OC=0
    X=0
    Y=0
    Energy=0
    COUNTLIST1=[]
    flag=COUNTLIST[0][0]
    for i in COUNTLIST:
        if cmp(i[0],flag)==0:
            if i[3]==i[4]:
                IC=IC+1
            else:
                OC=OC+1
            X=i[1]
            Y=i[2]
        else:
            if IC==0:
                Energy=10000
            else:
                Energy=float(OC)/IC
            COUNTLIST1.append([X,Y,Energy])
            IC=0
            OC=0
            flag=i[0]
```

```
SelectX1=[]
SelectX11=[]
SelectY1=[]
SelectY11=[]
L_COUNTLIST1=len(COUNTLIST1)
L1=int((L_COUNTLIST1)*0.5)
L2=L_COUNTLIST1-1-L1
COUNTLIST1.sort(key=operator.itemgetter(2))
COUNTLIST1.reverse()
for i in range(0,L1):
    SelectX1.append(COUNTLIST1[i][0])
    SelectY1.append(COUNTLIST1[i][1])
for i in range(0,L1):
    SelectX11.append(COUNTLIST1[i][0])
    SelectY11.append(COUNTLIST1[i][1])
Xlist=[]
Ylist=[]
Xlist.extend(SelectX1)
Xlist.extend(SelectX11)
Ylist.extend(SelectY1)
Ylist.extend(SelectY11)
xmax=float(heapq.nlargest(1,Xlist)[0])
xmin=float(heapq.nsmallest(1,Xlist)[0])
ymax=float(heapq.nlargest(1,Ylist)[0])
ymin=float(heapq.nsmallest(1,Ylist)[0])
OX=(xmax+xmin)/2
OY=(ymax+ymin)/2

R=(max(float(heapq.nlargest(1,Xlist)[0])-float(heapq.nsmallest(1,Xlist)[0]),float(
heapq.nlargest(1,Ylist)[0])-float(heapq.nsmallest(1,Ylist)[0])))/2
```

```
        for i in range(0,len(SelectX1)-1):
            Ratio=float(i)/L1
            sign1=0
            sign2=0
            if random.uniform(10,30)>20:
                sign1=1
                Dis=random.uniform(-0.2*R,R)*Ratio
                New_R=Dis+R
                SelectX1[i]=random.uniform(0,New_R)*sign1+OX
                if random.uniform(10,30)>20:
                    sign2=1

SelectY1[i]=OY+(math.sqrt(New_R**2-(SelectX1[i]-OX)**2))*sign2
                else:
                    sign2=-1

SelectY1[i]=OY+(math.sqrt(New_R**2-(SelectX1[i]-OX)**2))*sign2
            else:
                sign1=-1
                Dis=random.uniform(-0.2*R,R)*Ratio
                New_R=Dis+R
                SelectX1[i]=random.uniform(0,New_R)*sign1+OX
                if random.uniform(10,30)>20:
                    sign2=1
                    SelectY1[i]=OY+(math.sqrt(New_R                **
2-(SelectX1[i]-OX) ** 2))*sign2
                else:
                    sign2=-1
                    SelectY1[i]=OY+(math.sqrt(New_R                **
2-(SelectX1[i]-OX) ** 2))*sign2
```

```
        return SelectX1,SelectY1,SelectX11,SelectY11


    def Classify_Points(A):#Classify Points by Field1
        Xlist1=[]
        Xlist11=[]
        Ylist1=[]
        Ylist11=[]
        flag=A[0][3]
        COUNTLIST=[]
        for i in A:
            if cmp(i[3],flag)==0:
                COUNTLIST.append([i[0],i[1],i[2],i[3],i[6]])
            else:

SelectX1,SelectY1,SelectX2,SelectY2=Select_Points(COUNTLIST)
                Xlist1.extend(SelectX1)
                Xlist11.extend(SelectX2)
                Ylist1.extend(SelectY1)
                Ylist11.extend(SelectY2)
                flag=i[3]
                COUNTLIST=[]
                COUNTLIST.append([i[0],i[1],i[2],i[3],i[6]])
        SelectX1,SelectY1,SelectX11,SelectY11=Select_Points(COUNTLIST)
        Xlist1.extend(SelectX1)
        Xlist11.extend(SelectX11)
        Ylist1.extend(SelectY1)
        Ylist11.extend(SelectY11)
        return Xlist1,Ylist1,Xlist11,Ylist11
```

```
def main():
    csvfile_PSYCOUNT = file('PSYCOUNT.csv', 'rb')
    reader_PSYCOUNT = csv.reader(csvfile_PSYCOUNT)
    csvfile_PSY = file('Psychology.csv', 'rb')
    reader_PSY = csv.reader(csvfile_PSY)
    PSYCOUNT=[]
    SlectX1=[]
    SlectY1=[]
    SlectX2=[]
    SlectY2=[]
    Xlist1=[]
    Xlist2=[]
    Ylist1=[]
    Ylist2=[]
    Xlist=[]
    Ylist=[]
    Xlist0=[]
    Ylist0=[]
    for i in reader_PSYCOUNT:
        PSYCOUNT.append(i)
    del PSYCOUNT[0]

    for i in reader_PSY:
        Xlist0.append(i[1])
        Ylist0.append(i[2])
    del Xlist0[0]
    del Ylist0[0]
    # print PSYCOUNT[0]
    flag_Field1=PSYCOUNT[0][3]
    flag_PaperID=PSYCOUNT[0][0]
```

```
Xlist1,Ylist1,Xlist11,Ylist11=Classify_Points(PSYCOUNT)
Xlist.extend(Xlist1)
Xlist.extend(Xlist11)
Ylist.extend(Ylist1)
Ylist.extend(Ylist11)
xmax=heapq.nlargest(1,Xlist)[0]
xmin=heapq.nsmallest(1,Xlist)[0]
ymax=heapq.nlargest(1,Ylist)[0]
ymin=heapq.nsmallest(1,Ylist)[0]
print xmax,xmin,ymax,ymin
plt.title("Psychology")
plt.xlim(float(xmin)-20000,float(xmax)+20000)
plt.ylim(float(ymin)-20000,float(ymax)+20000)
plt.xlabel("x")
plt.ylabel("y")
Xlist2 =[]
Ylist2 =[]
print len(Xlist0)
for i in Xlist0:
    if i not in Xlist11:
        Xlist2.append(i)
for i in Ylist0:
    if i not in Ylist11:
        Ylist2.append(i)
print len(Xlist1),len(Ylist1),len(Xlist2),len(Ylist2)

plt.scatter(Xlist1,Ylist1,s=0.8,color='r')
```

```
            plt.scatter(Xlist2,Ylist2,s=0.8,color='m')
            # plt.scatter(Xlist,Ylist,s=0.1,color='r')
            # plt.scatter(Xlist,Ylist,s=0.2,color='m')
            plt.show()
            csvfile_PSYCOUNT.close()
            csvfile_PSY.close()

main()
```