# Homework2

## Q1

*Q1. (Implementation project) The DBLP dataset ([https://www.aminer.cn/citation](https://www.aminer.cn/citation)) consists of over one million entries of research papers published in computer science conferences and journals. Among these entries, there are a good number of authors that have coauthor relationships. Use a small dataset including 1,000 papers (can be downloaded from aminer), complete the following tasks.*

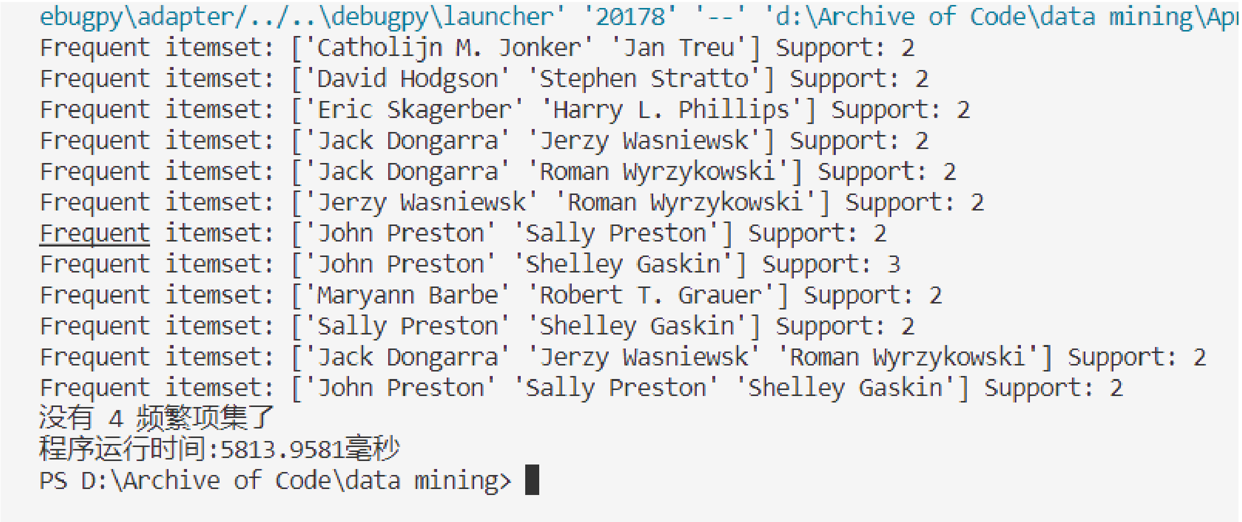
### (a) Propose a method that can mine efficiently a set of co-author relationships that are closely correlated (e.g., often co-authoring papers together).

修改Homework1的Apriori算法，使之能够挖掘论文作者之间合作关系的频繁模式。

主要修改部分包括：IO、频繁一项式的剪枝。

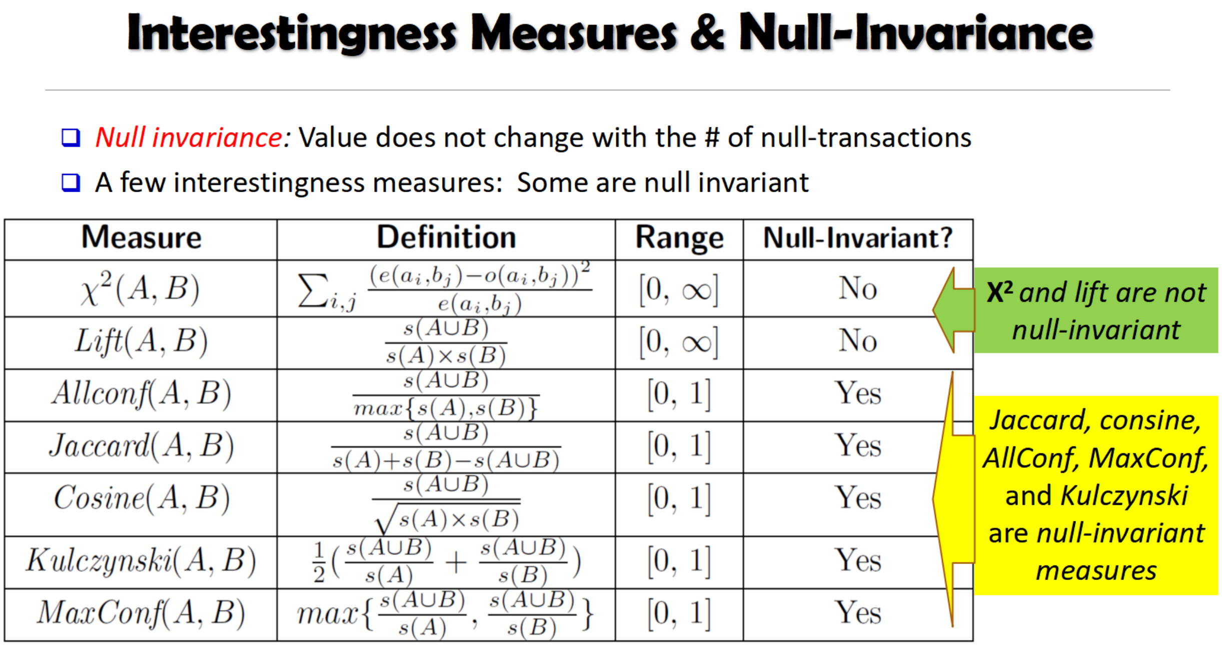
```
#######################################
#作业1读取的是int形数据，这里需要读取str数据#
#######################################
with open('outputacm.txt', 'r', encoding='utf-8') as f:
    for line in f:
        if line_count == max_line:
            break;
        if line != '\n' and line[1]=="@":
            # print(line)
            authors = line[2:-2]
            if authors == '':
                continue
            authors_list = authors.split(',')
            datalist.append(authors_list)
            flatdata = np.append(flatdata, authors_list)
            line_count += 1
            # paper_dict[line_count] = authors_list
        else:
            continue;


........................................


########################################################
#作业1中剪除非频繁一项式的方法是向量相乘，这里为str所以必须用遍历#
########################################################
unique, counts = np.unique(flatdata, return_counts=True)

flag = np.where(counts < support_rate * max_line, 0, 1)
one_frequent = unique[flag == 1]
one_frequent_counts = counts[flag == 1]
```

实验结果（**读取1000篇论文，支持度为0.002**）：



```
ebugpy\adapter/../..\debugpy\launcher' '20178' '--' 'd:\Archive of Code\data mining\Ap
Frequent itemset: ['Catholijn M. Jonker' 'Jan Treu'] Support: 2
Frequent itemset: ['David Hodgson' 'Stephen Stratto'] Support: 2
Frequent itemset: ['Eric Skagerber' 'Harry L. Phillips'] Support: 2
Frequent itemset: ['Jack Dongarra' 'Jerzy Wasniewsk'] Support: 2
Frequent itemset: ['Jack Dongarra' 'Roman Wyrzykowski'] Support: 2
Frequent itemset: ['Jerzy Wasniewsk' 'Roman Wyrzykowski'] Support: 2
Frequent itemset: ['John Preston' 'Sally Preston'] Support: 2
Frequent itemset: ['John Preston' 'Shelley Gaskin'] Support: 3
Frequent itemset: ['Maryann Barbe' 'Robert T. Grauer'] Support: 2
Frequent itemset: ['Sally Preston' 'Shelley Gaskin'] Support: 2
Frequent itemset: ['Jack Dongarra' 'Jerzy Wasniewsk' 'Roman Wyrzykowski'] Support: 2
Frequent itemset: ['John Preston' 'Sally Preston' 'Shelley Gaskin'] Support: 2
没有 4 频繁项集了
程序运行时间:5813.9581毫秒
PS D:\Archive of Code\data mining>
```

**(b) Based on the mining results and the pattern evaluation measures discussed in this chapter, discuss which measure may convincingly uncover close collaboration patterns better than others.**



## Interestingness Measures & Null-Invariance

- *Null invariance:* Value does not change with the # of null-transactions
- A few interestingness measures: Some are null invariant

| Measure | Definition | Range | Null-Invariant? |
|---|---|---|---|
| $\chi^2(A,B)$ | $\sum_{i,j}\frac{(e(a_i,b_j)-o(a_i,b_j))^2}{e(a_i,b_j)}$ | $[0,\infty]$ | No |
| $Lift(A,B)$ | $\frac{s(A\cup B)}{s(A)\times s(B)}$ | $[0,\infty]$ | No |
| $Allconf(A,B)$ | $\frac{s(A\cup B)}{max\{s(A),s(B)\}}$ | $[0,1]$ | Yes |
| $Jaccard(A,B)$ | $\frac{s(A\cup B)}{s(A)+s(B)-s(A\cup B)}$ | $[0,1]$ | Yes |
| $Cosine(A,B)$ | $\frac{s(A\cup B)}{\sqrt{s(A)\times s(B)}}$ | $[0,1]$ | Yes |
| $Kulczynski(A,B)$ | $\frac{1}{2}(\frac{s(A\cup B)}{s(A)}+\frac{s(A\cup B)}{s(B)})$ | $[0,1]$ | Yes |
| $MaxConf(A,B)$ | $max\{\frac{s(A\cup B)}{s(A)},\frac{s(A\cup B)}{s(B)}\}$ | $[0,1]$ | Yes |

*X² and lift are not null-invariant*

*Jaccard, consine, AllConf, MaxConf,* and *Kulczynski* are *null-invariant measures*

课上我们学习了**Chi-Squared，lift, allconf, jaccard, cosine, kulczynski, maxconf**几种模式评估方法。

- **卡方和lift：**这两个度量方法可以来看哪些作者之间的合作频率比预期的要高；比如，如果我们发现两个作者A和B经常一起发表论文，并且这种频率远远超过了他们独立合作的预期，那么他们就可能有强烈的合作关系。但是在实验中无法检测出这种明显的异常合作情况（因为读取的数据太少了hhh）。因此，这两个度量方法不适合本次实验。
- **全置信度和最大置信度：**用来度量作者之间的合作频率；例如，如果作者X和Y经常一起合作，而且每次作者X出现时，作者Y也会出现，那么它们之间合作关系比较强。数值越高表示第二/第三作者和第一个作者的合作关系越密切。
- **Jaccard, Cosine, Kaczynski：**这些方法可以告诉我们多个作者之间的合作重叠程度。数值越高表示他们之间的重叠程度越大，合作关系越紧密。例如，如果我们发现作者M和作者N经常与作者P一起合作，而且他们之间的合作重叠度非常高，那么我们可以说他们之间的合作关系非常紧密。

很显然卡方和lift不是最好方法，因为它们都不具有空不变性，考虑到我们的数据集多数的支持度都太低，所以卡方和lift不合适。

Cosine衡量的是两个向量之间的夹角，通过计算两个向量的内积与各自的模长之间的比例来度量相似性。在合作关系的背景下，我们可以将作者看作向量的维度，而他们之间的合作关系可以表示为向量之间的夹角。因此Cosine只关注向量的方向而忽略了向量的长度，因此在衡量合作关系的紧密程度时可能不够准确。

Kaczynski基于交集和并集大小之间的比例的相似度。但是和Jaccard相比Kaczynski对交集的大小更加敏感，反而忽略了并集的大小。在合作关系的背景下，我们更关注合作作者之间的共同合作程度，而不仅仅是合作关系的重叠程度。

所以**Jaccard**在几个方法中显然表现得更好。

### (c) Based on the study above, can you develop a method that can roughly predict advisor and advisee relationships and the approximate period for such advisory supervision?

这部分作业，我脑袋里的第一想法是用机器学习的方法，但是既然说只能用课上学到的方法，**我从Jaccard算法获得灵感，做出如下假设：**

- 首先，如果想推断两个人的指导与被指导关系，起码**需要很多包含两个人独立的或者合作的论文。**
- 其次，如果作者A和作者B**多次合作**，说明它们**可能存在指导与被指导关系。**
- 最后，如果**作者A的著作比作者B的著作多**，那么A更有可能是导师，B是学生。

然后我们可以简单写出以下伪代码：

```
paper_num = 3000;
min_support = 3;

one_frequent, one_frequent_support = find_frequent_one_item_set();
two_frequent, two_frequent_support = find_frequent_two_item_set(one_frequent,
one_frequent_support);
//这里不需要获取频繁n>2项集，因为只需要推测两者之间的关系
for(int i=0, i<len(two_frequent), i++):
    author_1 = two_frequent[i].get(0)
    author_2 = two_frequent[i].get(1)
    if(one_frequent_support[author_1] > one_frequent_support[author_2])
        #作者一可能为导师，作者二可能为学生
    if(one_frequent_support[author_1] < one_frequent_support[author_2])
        #作者二可能为导师，作者一可能为学生
    if(one_frequent_support[author_1] = one_frequent_support[author_2])
        #无法推断。
```

我们通过实验算出频繁一项集和频繁二项集：

```
####One_frequent
Frequent itemset: ( II) Support: 3
Frequent itemset: ( Jr) Support: 10
Frequent itemset: ( Jr.) Support: 7
Frequent itemset: (Bart G. Farka) Support: 4
Frequent itemset: (Behrouz A. Forouza) Support: 3
Frequent itemset: (Brian Cul) Support: 3
Frequent itemset: (C. A. Brebbia) Support: 3
Frequent itemset: (Catholijn M. Jonker) Support: 3
Frequent itemset: (Cay S. Horstman) Support: 8
Frequent itemset: (Charles J. Brook) Support: 5
Frequent itemset: (Cheryl R. Shroc) Support: 3
Frequent itemset: (Dan Birle) Support: 4
Frequent itemset: (Dan Gooki) Support: 3
Frequent itemset: (Dan Oj) Support: 4
Frequent itemset: (Dario Pescado) Support: 3
Frequent itemset: (David Blatne) Support: 3
Frequent itemset: (David D. Busc) Support: 3
Frequent itemset: (David Hodgson) Support: 3
Frequent itemset: (David Pogu) Support: 3
Frequent itemset: (Doug Wals) Support: 3
Frequent itemset: (Ed Titte) Support: 3
Frequent itemset: (Elliot B. Koffma) Support: 4
Frequent itemset: (Eric Mylona) Support: 4
Frequent itemset: (Fletcher Blac) Support: 3
Frequent itemset: (Gary B. Shelly) Support: 10
Frequent itemset: (Greg Krame) Support: 3
Frequent itemset: (Jan Treu) Support: 3
Frequent itemset: (Jean Andrew) Support: 3
Frequent itemset: (Jeffrey J. Quasne) Support: 3
Frequent itemset: (John Preston) Support: 3
Frequent itemset: (John Walkenbac) Support: 3
Frequent itemset: (Julia Case Bradle) Support: 3
Frequent itemset: (June Jamrich Parsons) Support: 6
Frequent itemset: (Keith Gemmel) Support: 4
Frequent itemset: (Kenneth C. Laudo) Support: 3
Frequent itemset: (Laura Parkinso) Support: 3
Frequent itemset: (Linda I. O'Lear) Support: 3
Frequent itemset: (Margaret Levine Young) Support: 3
Frequent itemset: (Maryann Barbe) Support: 5
Frequent itemset: (Michael Knigh) Support: 3
Frequent itemset: (Michael Mille) Support: 5
Frequent itemset: (Mikkel Aalan) Support: 3
Frequent itemset: (Naba Barkakat) Support: 4
Frequent itemset: (P. K. McBrid) Support: 3
Frequent itemset: (Pamela W. Adam) Support: 4
Frequent itemset: (Patricia A. Hoefle) Support: 3
Frequent itemset: (Robert Grauer) Support: 3
Frequent itemset: (Robert T. Grauer) Support: 6
Frequent itemset: (Ruth Mara) Support: 5
Frequent itemset: (Scott Kelb) Support: 7
Frequent itemset: (Shelley Gaskin) Support: 6
Frequent itemset: (Staf) Support: 10
Frequent itemset: (Stanley Habi) Support: 5
Frequent itemset: (Stephen Stratto) Support: 4
Frequent itemset: (Steve Johnso) Support: 4
Frequent itemset: (Ted LoCasci) Support: 5
Frequent itemset: (Terry Sanche) Support: 3
Frequent itemset: (Thomas J. Cashman) Support: 10
Frequent itemset: (Tim Bogen) Support: 4
Frequent itemset: (Timothy J. O'Leary) Support: 3
Frequent itemset: (Tony Johnso) Support: 3
Frequent itemset: (Walter Savitc) Support: 3
Frequent itemset: (Yashavant P. Kanetka) Support: 5
#### Two _frequent
Frequent itemset: ['Catholijn M. Jonker' 'Jan Treu'] Support: 3
Frequent itemset: ['Dan Oj' 'June Jamrich Parsons'] Support: 4
Frequent itemset: ['Gary B. Shelly' 'Jeffrey J. Quasne'] Support: 3
Frequent itemset: ['Gary B. Shelly' 'Thomas J. Cashman'] Support: 10
Frequent itemset: ['Jeffrey J. Quasne' 'Thomas J. Cashman'] Support: 3
Frequent itemset: ['John Preston' 'Shelley Gaskin'] Support: 3
Frequent itemset: ["Linda I. O'Lear" "Timothy J. O'Leary"] Support: 3
Frequent itemset: ['Maryann Barbe' 'Robert T. Grauer'] Support: 5
#### 3 _frequent
Frequent itemset: ['Gary B. Shelly' 'Jeffrey J. Quasne' 'Thomas J. Cashman']
Support: 3
```

**可以发现，二项集['Dan Oj' 'June Jamrich Parsons']中，'Dan Oj'支持度为3，'June Jamrich Parsons'支持度为6，所以通过我们的假设，'June Jamrich Parsons'可能是'Dan Oj'的导师，其他类似**

# Code Appendix

见附录。