基于关联规则挖掘的学者关系分析框架

摘要

随着网络技术的高速发展，海量信息不断涌现。然而，过多信息导致难以提炼有用知识，造成“信息丰富而知识贫乏”的窘境。因此，人们迫切希望能对海量数据进行深入分析，发现并提取隐藏在其中的知识，这就是数据挖掘（Data Mining）。本文提出了一种学者关系分析框架，通过对发表情况进行关联规则挖掘，提取合著者、团队等学者关系，并分析学者关系的活跃程度。在实验部分，本文使用 FP-Growth 算法，基于 DBLP 数据集，从以下三方面进行分析：1）发文数量变化趋势，2）活跃学者结构特征，3）活跃学者变化趋势。代码已开源：https://github.com/MoonOutCloudBack/DBLP-mining。

Abstract

With the rapid development of Informational Technology, massive information is constantly emerging. Nevertheless, too much information makes it difficult to extract useful knowledge, resulting in the dilemma of "rich information while poor knowledge". Thus, people are eager to create an approach to analyze the massive data and extract hidden knowledge, which is called Data Mining. In this paper, we present a framework for analyzing scholar relationships. By mining the Association Rules of scholars' publications, we extract the scholar relationships such as co-authors and teams, as well as analyzes the activity of these relationships. In experiments, we use FP-growth algorithm to analyze the DBLP dataset from the following three points of view: 1) the trend of the publications numbers, 2) the structural characteristics of active scholars, and 3) the trend of active scholars. The source code is open-sourced at https://github.com/MoonOutCloudBack/DBLP-mining.

一、关联规则挖掘

1.1 基本定义

关联规则（Association Rules），反映了一个事物与其他事物之间的相互依存性和关联性。关联规则挖掘是数据挖掘的一个重要技术，用于从大量数据中挖掘出有价值的数据项之间的相关关系，可从数据库中分析出形如“由于某些事件的发生而引起另外一些事件的发生”之类的规则。[1]

项集，表示在数据集中一起出现的元素的集合。频繁项集，表示在数据集中频繁出

现的项集，他们可能具有关联关系。频繁的判定方式、以及讲过的评估标准，有支持度，置信度和提升度三个。[1]

- 支持度：几个关联的数据在数据集中出现的次数，占总数据集的比例。

$$\text{Support}(X,Y) = P(XY) = \frac{\text{num}(XY)}{\text{num}(\text{AllSamples})}$$

- 置信度：一个数据出现后另一个数据出现的概率，即数据的条件概率。

$$\text{Confidence}(X|Y) = P(X|Y) = \frac{P(XY)}{P(Y)}$$

- 提升度：表示含有 Y 的条件下同时含有 X 的概率，与 X 总体发生的概率之比，即含有 Y 能使含有 X 的概率得到多大提升。

$$\text{Lift}(X|Y) = \frac{P(X|Y)}{P(Y)} = \frac{\text{Confindence}(X|Y)}{P(X)}$$

1.2 FP-Growth 算法

FP-growth（Frequent Pattern Tree，频繁模式树）是 Jiawei Han 提出的关联规则挖掘方法[2]，使用 FP-tree 数据结构存储数据集，在整个算法执行过程中，只需遍历数据集 2 次，就能够完成频繁模式发现，比 Apriori 算法效率更高。[5]

其发现频繁项集的基本过程如下：

（1）构建 FP 树；

（2）从 FP 树中挖掘频繁项集

此处列出几篇参考博客，将 FP-Growth 的原理解释很清楚。

- [3]：FP-growth 算法与 Python 实现，https://blog.csdn.net/songbinxu/article/details/80411388
- [4]：FP-growth 算法原理及 python 实现（详细代码解释），https://blog.csdn.net/weixin_42419314/article/details/83617684
- [5]：FP-growth 算法理解和实现，https://blog.csdn.net/baixiangxue/article/details/80335469

二、基于关联规则挖掘的学者关系分析框架

2.1 学者关系：合著者、团队

我们设置支持度阈值 s 与置信度阈值 c。寻找这样的 2 个及以上元素：支持度大于 s，且任一元素与除它之外其他元素的置信度大于 c。认为这些元素组成的项集为有效学者关系。其中，2 人的学者关系为合著者，2 人以上的学者关系为团队。

2.2 分析学者关系的活跃程度

认为学者关系的活跃程度，与学者关系发表文章数量、学者关系发表文章数量与学者总发表文章数量的比值，呈正相关。设目前有学者关系 $R = (a_1, a_2, \ldots, a_n)$ ，num(R)表示学者关系发表文章数量，num(a)表示学者发表文章数量。定义活跃程度如下：

$$\mathrm{Active(R)} = \alpha \mathrm{num(R)} + \frac{\beta}{n}\sum_{i=1}^{n}\frac{num(R)}{num(a_i)}$$

其中 α, β 均为大于 0 的实数。

三、基于 FP-Growth 算法的 DBLP 数据集分析

3.1 实验参数设置

自从 2017 年论文“Attention is all you need”发表，使用 Transformer 结构和 Attention 结构的深度学习工作不断涌现。我们用关键词“Attention”和“Transformer”对发表文章进行筛选，同时按年对发表情况进行分析。设 1. 挖掘合著者、团队关系的支持度阈值 $s = \frac{5}{\text{该年总发表数量}}$、置信度阈值 $c = 0.5$；2. 活跃程度计算中 $\alpha = 1$、$\beta = 10$。

3.2 合著者与团队

下表列出了每年合著者与团队数量：

表 3-1 每年合著者与团队数量

年份	合著者数量	团队数量
2017	9	0
2018	28	15
2019	85	51
2020	116	60
2021	223	124
2022	8	5

可以得到以下结论：

1. 对 Attention 和 Transformer 的研究数量不断增多，分别在 2019、2021 年激增。
2. 两人合作的情况比团队研究更多，数量关系上大致多一倍。

3.3 学者关系活跃程度

下列表格列出了每年活跃程度前 5 的合著者/团队的成员、发表数量、活跃程度：

表 3-2 2017-2022 年活跃程度前 5 的合著者

年份	成员	发表数量	活跃程度
2017	('haewoon kwak', 'jisun an')	6	37.5
	('yutaka matsuo', 'edison marrese-taylor')	6	37.09091
	('zhou zhao', 'yueting zhuang')	6	36.4492
	('xiaogang wang', 'wanli ouyang')	6	36.44788
	('hanwang zhang', 'tat-seng chua')	4	24.56863
2018	('siu cheung hui', 'yi tay')	13	79.2309
	('siu cheung hui', 'luu anh tuan')	7	43.5
	('yi tay', 'luu anh tuan')	7	43.16279
	('zhe lin', 'xiaohui shen')	6	37.16667
	('anh tuan luu', 'yi tay')	6	36.99668
2019	('zhaopeng tu', 'xing wang')	12	73.15294
	('zhaopeng tu', 'baosong yang')	10	60.79412
	('lidia s. chao', 'derek f. wong')	9	55.24286
	('fangzhao wu', 'xing xie')	9	54.875
	('bj', 'rn w. schuller')	9	54.4564
2020	('bj', 'rn w. schuller')	11	66.55782
	('thomas hain', 'qiang huang')	9	55.0625
	('yongqiang wang', 'frank zhang')	9	55.01786
	('alessandro mingotti', 'lorenzo peretto')	8	49.06667
	('clinton fookes', 'simon denman')	8	48.89164
2021	(' j', 'gou')	12	73.55
	('herv', 'gou')	12	73.46667
	('herv', ' j')	12	73.41667
	('xiyang dai', 'lu yuan')	11	67.43277
	('wengang zhou', 'houqiang li')	10	61.17647
2022	('long yu', 'shengwei tian')	6	36.68627
	('kailun yang', 'rainer stiefelhagen')	6	36.73333
	('jiaming zhang', 'rainer stiefelhagen')	6	36.82857
	('jiaming zhang', 'kailun yang')	6	36.7619
	('jiaming zhang', 'kunyu peng')	5	30.9127

表 3-3 2017-2022 年活跃程度前 5 的团队

年份	成员	发表数量	活跃程度
2018	('siu cheung hui', 'yi tay', 'luu anh tuan')	7	31.99612
	('tianyi zhou', 'chengqi zhang', 'tao shen')	6	27.15233
	('tianyi zhou', 'chengqi zhang', 'guodong long')	6	27.04344
	('tianyi zhou', 'guodong long', 'tao shen')	6	26.98422
	('tianyi zhou', 'jing jiang', 'tao shen')	6	26.93949
2019	('zhizhong han', 'matthias zwicker', 'yu-shen liu')	7	31.85686
	('lidia s. chao', 'baosong yang', 'derek f. wong')	7	31.65
	('albert zeyer', 'ralf schl', 'ter')	7	31.57222
	('albert zeyer', 'ralf schl', 'hermann ney')	7	31.4636
	('chuhan wu', 'fangzhao wu', 'xing xie')	7	31.34722
2020	('chunyang wu', 'ching-feng yeh', 'yangyang shi')	7	32.13636
	('chunyang wu', 'ching-feng yeh', 'frank zhang')	7	32.05303
	('ching-feng yeh', 'yangyang shi', 'frank zhang')	7	32.05303
	('gui-bin bian', 'xiao-liang xie', 'zeng-guang hou')	7	32.04861
	('chunyang wu', 'yangyang shi', 'frank zhang')	7	32
2021	('herv', ' j', 'gou')	12	54.21667
	('weijian xu', 'yifan xu', 'zhuowen tu')	9	41.41818
	(' j', 'gou', 'hugo touvron')	9	40.85481
	('herv', 'gou', 'hugo touvron')	9	40.79231
	('herv', ' j', 'hugo touvron')	9	40.75481
2022	('jiaming zhang', 'kailun yang', 'rainer stiefelhagen')	6	27.1619
	('jiaming zhang', 'kunyu peng', 'rainer stiefelhagen')	5	22.9127
	('jiaming zhang', 'kailun yang', 'kunyu peng')	5	22.85714
	('kailun yang', 'kunyu peng', 'rainer stiefelhagen')	5	22.83333
	('jiaming zhang', 'kailun yang', 'kunyu peng', 'rainer stiefelhagen')	5	19.02381

可以得到以下结论：

1. 活跃团队基本都是三位学者组成，很少出现 4 位以上学者。
2. 活跃的团队里，成员经常相互重复；推测他们出自一个实验室或一个学院，可能符合“两个固定导师+一个可变学生”、“三个相互熟悉的导师”等三人合作模式。

3. 活跃的合著者里，成员也经常互相重复；最常出现的是“固定 A+可变 B,C,D”合著者模式，推测可能为“一个固定导师+一个可变学生”合作模式；甚至出现 ABC 三人中，任意两两组合都是活跃合作者的情况（如 2021、2022 年）。
4. 在 2018 年，最活跃合著者的发表数量激增（13），比 2017 年多一倍（6），此后一直维持该数量（11-13）。
5. 在 2021 年，最活跃团队的发表数量激增（13），比以往多一倍（7）。
6. 观察人名发现，华人学者占了相当大比例。
7. 活跃合著者/团队更替相当快，没有发现合著者/团队在这一领域保持高度活跃超过一年。

四、结论与展望

本文提出了一种学者关系分析框架：通过对发表情况进行关联规则挖掘，提取合著者、团队等学者关系；定义了学者关系的活跃程度，对学者关系进行有效分析。在实验部分，本文使用 FP-Growth 算法，基于 DBLP 数据集，进行具体分析，证明了该框架的可行性，提炼出了有效的学者关系。

对于未来工作，可以考虑以下方向：

- 在评判学者发表情况、学者关系时，将文章质量纳入考虑，如按照被引数给文章加权；需要借助 DBLP 外的外部数据。
- 对部分学者关系，添加可视化网络图；进一步，可以考虑使用图的数据结构进行数据挖掘。
- 对活跃学者的发表文章题目进行数据挖掘，提炼出题目中重复出现的关键词。

参考文献

[1] 关联规则（Association Rules）学习，https://blog.csdn.net/weixin_40042143/article/details/82691106

[2] Han, Jiawei, et al. "Mining frequent patterns without candidate generation: A frequent-pattern tree approach." Data mining and knowledge discovery 8.1 (2004): 53-87.

[3] FP-growth 算法与 Python 实现，https://blog.csdn.net/songbinxu/article/details/80411388

[4] FP-growth 算法原理及 python 实现（详细代码解释），https://blog.csdn.net/weixin_42419314/article/details/83617684

[5] FP-growth 算法理解和实现，https://blog.csdn.net/baixiangxue/article/details/80335469