2018年10月29日星期一

上午9时22分

从用户人格特质与组织关系的角度建模其人际关系，从而进一步与离职用户交叉匹配，建模其工作满意度的动态变化。

本部分的理论假设为：

1. 都说“物以类聚，人以群分”，加上反生产行为中的CPB-I涉及到同事间人际处理特点，因此有理由相信人格相似的人更容易走近，其CPB-I都比较低，从而更容易相处；即CPB-I特征越相近，走到一起的概率越大；
2. 人际发生效应的媒介是日常接触，而日常接触的机会很大程度上取决于组织架构，如同一个团队，同一个部门，或者不同部门的相同职位（需要交流等），因此决定对于用户的组织结构进行标记

接下来的第一个问题是如何对于用户的组织信息进行标记，具体有两类：

* 假设不同父组织下的同名子组织标号相同，如执行1部与执行2部下各自的管理1组，分别采用01与11表示，第一位表示部门，第二位表示组号；此种情况将不同组织下的同名组认为是相同的进行聚类；
* 认为不同组织下的小组等完全不同，重新标记，即采用累加标记方法，该方法首先对每个部门书下的层数分别标记，然后将标记过后的数值累加得到该组织标号，如此不同组织下的子组织标号完全不同，如上述例子，各自管理1组的标号分别为0+0=0与1+0=1

例子：

['1 - Executive', '2 - ResearchAndEngineering', '2 - Research', '1 - Lab']

['2 - Executive', '2 - ResearchAndEngineering', '2 - Research', '1 - Lab']

上述两个’1 - Lab’是否不同的呢？我们的初始假设是因为不同的，因为比如分处北京与济南两地事业部下，即便都是研发团队，交流也并不多，更多取决于不同的业务场景与区域特点；甚至还可能是竞争关系。因此我们默认假设二者是不同的。

因此一种方案是：

我们将不同的组织名称模块化，即不同组织中的相同模块看作是同一的，其不同取决于上下文，即整体的组织标号序列，且：

* 分层级统计不同模块下的组织标号；
* 默认0表示该层级没有模块，起始模块从1-N
* 组织标号序列反映了用户所处的组织中的位置

还有一种方案则是：

采用累加进制的形式加剧结构差异的影响，即：

* 每层默认标号0表示缺失
* 每层的标号是之前层次标号的和+本层组织排序标号
* 本层组织标号的方案依旧是全组织内部同层的编号

我们先按照此类方法开始聚类分析，如遇到问题，再考虑其他的编码方案。

因此，我们现在0.3实验的第一个子实验为：

实验1：基于用户OCEAN与OS结构的人际关系建模，以及在此基础上的JS识别

本日10月29日必须完成上述实验1

实验大致步骤：

1. 获取用户的OCEAN人格特质分数
2. 获取用户的周围离职人数关系
3. 对CERT5.2的LDAP组织进行编码，形成四维度特征；
4. 用户的OCEAN特征（3维度或者5维度）+四维度OS特征==7维度的人际关系特征；KMeans聚类，得到所有2000个用户聚类结果，以及分别两个事业部的分别聚类结果
5. 分别从企业整体2000个用户以及分事业部两个层次对于用户的人际关系进行圈定，然后与用户的离职人数进行交叉匹配，记录其出现的离职人数个数，以此作为原始的JS\_Risk标志；
6. 所有2000个用户或者分事业部分别依据JS\_Risk排序，查看场景二中Insiders的位置是否合理
7. 依据发现的问题继续研究思考，改进

2018年10月30日星期二

上午9时26分

由于涉及到KMeans的参数K的上限问题，故首先尝试使用DBSCAN进行密度聚类，希望可以发现合适的参数；

上午自己利用一个小时的时间使用密度聚类DBSCAN进行了实验，采用eps=[0,1]步长0.02遍历，结果最好的轮廓系数才0.26，不得不说太低了。因此直接放弃。

考虑是因为数据较为稀疏导致的。

继续进行KMeans聚类，关键是初始K的确定。对于1000-2000多人的大企业组织而言，K=10显然是不够的，我们这里默认采取的默认K上限是所有团队的总数，一方面我们在台式机上进行2000个用户的143上限的聚类，一方面我们在笔记本上分事业部进行聚类分析。

统计30个场景二攻击用户的事业部分布为：1事业部20人，2事业部10人

属于第一事业部的攻击二有： 20

VCF1602

CKP0630

ITA0159

ICB1354

RRS0056

SIS0042

CIF1430

HMS1658

BYO1846

KSS1005

GWG0497

TRC1838

TNB1616

HIS1394

MCP0611

NAH1366

LVF1626

WDT1634

HXP0976

OKM1092

属于第二事业部的攻击二有： 10

DCC1119

MDS0680

CHP1711

OSS1463

JAL0811

HSN0675

ZIE0741

SNK1280

TMT0851

MGB1235

2018年10月30日星期二

上午11时16分

第一次KMeans实验：针对Part1的事业部1用户而言，最佳聚类竟然是：

....<<<<对于KMeans而言，最佳K聚类为： >>>>.... （MinMax处理）

2 0.23370773857895727

....<<<<对于KMeans而言，最佳K聚类为： >>>>.... （scale处理）

2 0.247966149014729

2018年10月30日星期二

下午4时18分

经过上午简单的实验，发现了一个严重的问题：

无论是对于CERT5.2的All来说，还是对于Part1/Part2而言，依据Big5+OS编码的特征聚类，无论是使用KMeans或者DBSCAN竟然最好的都是2个类别！

让我静静！

如果单独考虑Part1中的OCEAN聚类，则

....<<<<对于KMeans而言，最佳K聚类为： >>>>....

14 0.2541766404971729