



中山大學
SUN YAT-SEN UNIVERSITY

本科生毕业论文（设计）

Undergraduate Graduation Thesis (Design)

题目 Title: 社交网络中异常用户的检测

院系
School (Department): 数据科学与计算机学院

专业
Major: 计算机科学与技术

学生姓名
Student Name: 张佐奇

学号
Student No.: 13349160

指导教师(职称)
Supervisor (Title): 吴嘉婧（讲师）

时间: 2017 年 05 月 01 日
Date: Month 05 Day 01 Year 2017

说 明

1. 毕业论文（设计）的写作格式要求请参照《中山大学本科生毕业论文的有关规定》和《中山大学本科生毕业论文（设计）写作与印制规范》。
2. 除完成毕业论文（设计）外，还须填写三份表格：
 - （1）表一 毕业论文（设计）开题报告；
 - （2）表二 毕业论文（设计）过程检查情况记录表；
 - （3）表三 毕业论文（设计）答辩情况。
3. 上述表格均可从教务部主页的“下载中心”处下载，如表格篇幅不够，可另附纸。每份毕业论文（设计）定稿装订时应随同附上这三份表格。
4. 封三是毕业论文（设计）成绩评定的主要依据，请认真填写。

Instruction

1. Please refer to '*The Guidelines to Undergraduate Graduation Thesis (Design) at Sun Yat-sen University*' and '*The Writing and Printing Format of Undergraduate Graduation Thesis(Design) at Sun Yat-sen University*' for anything about the thesis format.
2. Three forms should be filled up before the submission of the thesis (design) :
 - （1）Form 1: Research Proposal of Graduation Thesis.
 - （2）Form 2: Process Check-up Form.
 - （3）Form 3: Thesis Defense Performance Form.
3. All the above forms could be downloaded on the website of the Office of Education Administration. If there is not enough space in the form, please add extra sheets. Each thesis (design) should be submitted together with the three forms.
4. The form on the inside back cover is the grading sheet. Please fill it up before submission.

表一：毕业论文（设计）开题报告
Form 1: Research Proposal of Graduation Thesis (Design)

论文（设计）题目 社交网络中异常用户的检测 Thesis (Design) Title:
<p>（简述选题的目的、思路、方法、相关支持条件及进度安排等） （ Please briefly state the research objective, research methodology, research procedure and research schedule in this part. ）</p> <p>现如今社交网络已经成为人们生活、工作、交流的重要平台，然而在带给人们各种便利、满足人们各项需求的同时，其海量的用户数也吸引了攻击者，成为攻击者获取巨大利益的新乐园。攻击者通过创建大量的虚假账号和盗用正常用户的账号，在社交网站中发布广告、钓鱼等恶意信息。这些虚假账号和被盗用的账号统称为异常账号，异常账号的存在严重威胁了正常用户的信息安全和社交网络的信用体系，为此异常账号检测成为了当今社交网络安全研究的关键问题之一，目前有大量的研究工作来检测社交网络中异常账号。</p> <p>对于像我一样每天都会使用这些社交软件的人来说，这些异常用户的存在已经严重影响了正常使用，作为一个计算机科学专业的学生，如果能够利用自己的专业知识设计出一个检测社交网络异常用户的算法，并通过编程将其实现，相信一定能够保障我们的信息安全，提升用户的使用体验。目前实验室已经有大量的关于社交网络的数据可以用来分析，包括用户信息、用户关系以及用户行为等，从而为这个题目提供了很好的支持条件。学术领域内也已经有部分这方面的论文，通过对这些国内外论文的阅读，可以总结出一些合理的对异常用户的定义以及有效的检测算法，之后在这些算法的基础之上做出改进并通过编程实现，最后再通过已有的社交网络数据对检测方案进行验证。</p>

具体的进度安排如下：

2016 年 12 月：进行相关论文的阅读工作，总结异常用户的检测算法。

2017 年 1 月～2 月：提出改进的思路并设计算法。

2017 年 3 月：使用 C++编程实现算法。

2017 年 4 月：利用社交网络数据检测算法的有效性并改进。

Student Signature:

Date:

指导教师意见

Comments from Supervisor:

1.同意开题

2.修改后开题

3.重新开题

1.Approved()

2. Approved after Revision ()

3. Disapproved()

Supervisor Signature:

Date:

表二：毕业论文（设计）过程检查情况记录表
Form 2 : Process Check-up Form

指导教师分阶段检查论文的进展情况（要求过程检查记录不少于 3 次）

The supervisor should check up the working process for the thesis (design) and fill up the following check-up log. At least three times of the check-up should be done and kept on the log.

第 1 次检查（First Check-up）：

学生总结

Student Self-summary:

完成以下论文的阅读工作，并总结异常用户的定义以及检测算法。

- (1) A Probabilistic Approach to Uncovering Attributed Graph Anomalies
- (2) Discovering Structural Anomalies in Graph-based Data
- (3) Focused Clustering and Outlier Detection in Large Attributed Graphs
- (4) Graph-Based Anomaly Detection
- (5) Intrusion as Anti(social) Communication: Characterization and Detection

指导教师意见

Comments of Supervisor:

认真阅读几篇论文之后，尝试提出改进的算法思路。

第 2 次检查（Second Check-up）：

学生总结

Student Self-summary:

提出主要以图结构作为特征进行异常检测，学习 Network Embedding 三种算法，并编译开源代码尝试运行。寻找公开社交网络数据集，了解斯坦福大学网络分析平台 SNAP，尝试编写爬虫爬取新浪微博用户关注粉丝数据。

指导教师意见

Comments of Supervisor:

进度稍慢，需要尽快提出自己的解决方案。

第 3 次检查 (Third Check-up) :

学生总结

Student Self-summary:

提出基于 Network Embedding 的异常用户检测方案，生成节点图表征向量后进行聚类，认为数量较少的社区内的节点为异常节点。

指导教师意见

Comments of Supervisor:

论文内容比较初步，理论较少，方案不够具有说服力，需要首先解决对异常用户定义的问题。

第 4 次检查

Fourth Check-up

学生总结

Student Self-summary:

<p>修改之前的方案，提出连接多个社区的节点为异常用户的定义，在获得节点特征向量后，将网络分为若干个社区，设计目标函数学习节点属于每个社区的概率，因为异常节点属于不同社区的概率的分布倾向于比较平均，从而找出异常节点，最后使用模块度对结果进行检验。</p> <p>指导教师意见（Comments of Supervisor）： 改进较大，比之前内容上丰富了很多。</p> <p>学生签名（Student Signature）：日期（Date）：</p> <p>指导教师签名（Supervisor Signature）：日期（Date）：</p>	
<p>总体完成情况 (Overall Assessment)</p>	<p>指导教师意见 Comments of Supervisor:</p> <p>1、按计划完成，完成情况优（Excellent）： （ ） 2、按计划完成，完成情况良（Good）： （ ） 3、基本按计划完成，完成情况合格（Fair）： （ ） 4、完成情况不合格（Poor）： （ ）</p> <p>指导教师签名（Supervisor Signature）： 日期（Date）：</p>

表三：毕业论文（设计）答辩情况登记表
Form 3: Thesis Defense Performance Form

答辩人 Student Name		专 业 Major	
论文（设计）题目 Thesis（Design） Title			
答辩小组成员 Committee Members			
答辩记录 Records of Defense Performance:			
记录人签名（Clerk Signature）：		日期（Date）：	

学术诚信声明

本人所提交的毕业论文，是在导师的指导下，独立进行研究工作所取得的成果，所有数据、图片资料均真实可靠。除文中已经注明引用的内容外，本论文不包含任何其他人或集体已经发表或撰写过的作品或成果。对本论文的研究作出重要贡献的个人和集体，均已在文中以明确的方式标明。本毕业论文的知识产权归属于培养单位。本人完全意识到本声明的法律结果由本人承担。

本人签名：

日期：

Statement of Academic Integrity

I hereby acknowledge that the thesis submitted is a product of my own independent research under the supervision of my supervisor, and that all the data, statistics, pictures and materials are reliable and trustworthy, and that all the previous research and sources are appropriately marked in the thesis, and that the intellectual property of the thesis belongs to the school. I am fully aware of the legal effect of this statement.

Student Signature:

Date:

【摘要】

近几年，越来越多的人开始使用社交网络，社交网络成为了人们生活、工作、交流的重要组成部分。然而，社交网络在带给人们各种便利、满足人们各项需求的同时，其海量的用户数也吸引了许多的攻击者。攻击者通过异常账号在社交网络中发布大量的广告或非法信息，对社交网络中的正常用户的信息安全和使用体验造成了严重威胁。

本文首先提出了一种对于异常用户的合理的定义，然后在此基础上提出了基于图向量表征的异常用户检测算法。通过 Network Embedding 算法以及度的计算得到所有节点的特征向量，然后将网络分为若干个社区，为每一个节点学习一个隶属度向量，向量的每一维表示节点属于社区的概率，最后通过异常用户与正常用户之间隶属度向量的值分布的区别，找出存在于社交网络中的异常用户。

本文使用了安然公司内部员工邮件往来网络的真实数据集对提出的检测方案进行了实验，利用网络的模块度的变化验证算法的有效性，并得到了预期的较好的结果。在社区数量的三种不同取值下，删除算法检测出的异常节点后，网络的模块度分别提升了 2.9%、2.0%、3.2%。

【关键词】

社交网络；异常检测；社区发现；图向量表征

[ABSTRACT]

In recent years, more and more people start to use social network, social network has become one of the most important parts of people's daily life, work and communication. Social network has brought so much convenience and has fulfilled all kinds of people's demands, however, at the same time, the great amount of social network has also attracted many web attackers. They use abnormal accounts to publish lots of ads and illegal information in social network, which has posed great threat to the information security and user experience of other normal users.

This paper first comes up with a reasonable definition of anomaly user in social network, and then proposes an algorithm of anomaly detection using network embedding based on this definition. We use network embedding algorithm and calculation of degree to obtain the feature vector of each node in the network. Then we divide the network to several communities, each node is assigned a vector of degree of membership by our learning algorithm, each dimension of this vector represents the probability that the node belongs to a community. In this way, anomaly user can be detected by the difference of value distribution between its membership vector and others.

Dataset of the E-mail network of Enron Inc., a real-life social network dataset, is used for analysis, we use modularity to validate the efficiency of our proposed algorithm, and has yielded an expected result. For the three different values of the number of clusters, after deleting the anomaly nodes we have detected, the modularity of the network has increased 2.9%, 2.0% and 3.2%, respectively.

[Keywords]

Social Network; Anomaly Detection; Community Detection; Network Embedding

目录

第 1 章 引言.....	- 1 -
1.1 课题的背景和意义.....	- 1 -
1.2 课题的核心问题.....	- 3 -
1.2.1 异常用户的表现	- 3 -
1.2.2 检测方案的设计	- 4 -
1.2.3 检测方案的验证	- 4 -
1.3 课题的主要挑战.....	- 4 -
1.3.1 社交网络巨大的用户数据	- 4 -
1.3.2 异常用户的多种表现形式	- 4 -
1.3.3 异常用户特征的动态变化	- 5 -
1.4 课题的研究现状.....	- 5 -
1.4.1 基于行为特征的检测方案	- 6 -
1.4.2 基于内容的检测方案	- 6 -
1.4.3 基于图的检测方案	- 7 -
1.4.4 无监督学习的检测方案	- 8 -
1.5 本文的工作.....	- 8 -
1.6 论文结构简介.....	- 9 -
第 2 章 异常用户检测算法综述.....	- 11 -
2.1 基于全局统计规律的检测算法.....	- 11 -
2.2 基于图的局部结构的检测算法.....	- 12 -
2.3 基于提取特征的训练的检测算法.....	- 13 -
第 3 章 图论相关知识与异常用户定义.....	- 15 -
3.1 图论的相关知识.....	- 15 -
3.1.1 图	- 15 -
3.1.2 度	- 15 -
3.1.3 社区	- 15 -
3.2 异常用户的定义.....	- 15 -
第 4 章 基于图向量表征的异常用户检测算法.....	- 19 -
4.1 Network Embedding.....	- 19 -
4.1.1 DeepWalk	- 20 -

4.1.2	LINE	- 22 -
4.1.3	node2vec	- 23 -
4.2	基于图向量表征的异常用户检测算法	- 26 -
4.2.1	特征的提取	- 26 -
4.2.2	目标函数的设计	- 26 -
4.2.3	目标函数的求解	- 28 -
4.2.4	隶属度向量的学习	- 30 -
4.2.5	异常用户的判断	- 31 -
第 5 章	实验过程与结果分析	- 33 -
5.1	数据集的获取	- 33 -
5.1.1	爬虫获取	- 33 -
5.1.2	公开数据集	- 33 -
5.2	数据集的描述	- 33 -
5.3	实验过程	- 34 -
5.3.1	生成节点特征向量	- 34 -
5.3.2	初始化节点聚类	- 36 -
5.3.3	学习隶属度向量	- 37 -
5.3.4	判断异常用户	- 37 -
5.4	实验结果与分析	- 37 -
第 6 章	结语	- 41 -
参考文献:		- 43 -
致谢		- 46 -

第1章 引言

1.1 课题的背景和意义

社交网络即社交网络服务（Social Network Service, SNS），是指为拥有共同兴趣、行为、背景的人们建立社交关系的在线网络平台^[13]。随着 Internet 用户的普及以及 Web 2.0 技术的成熟，社交网络呈现出飞速发展的趋势。



图 1-1：社交网站的种类繁多

如图 1-1 所示，越来越多的社交网站出现在我们的生活中，其中包括国外的脸书（Facebook）、推特（Twitter）、领英（LinkedIn），国内的新浪微博、腾讯微博、人人网等，这些社交网站都聚集了大量的用户。据统计¹，截止至 2016 年 12 月 31 日，Facebook 已经拥有来自全世界的 18.6 亿活跃用户（如图 1-2 所示^[10]），国内的新浪微博的用户注册数也已超过 5 亿。

¹ https://en.wikipedia.org/wiki/List_of_virtual_communities_with_more_than_100_million_active_users



图 1-2：Facebook 拥有来自世界各地的海量用户

社交网络如今已经成为人们生活、工作、交流的重要平台，用户们不仅把现实生活中的人际关系搬到了社交网络上，还建立了与线下无关的单纯线上朋友关系，在社交网络上搭建起全新的沟通和分享信息的平台。

然而，社交网络在带给人们各种便利、满足人们各项需求的同时，其海量的用户数也吸引了许多的攻击者^{[15][14][7]}，成为了攻击者获取巨大利益的新乐园。攻击者通过创建大量的虚假账号和盗用正常用户的账号，在社交网络中发布广告、钓鱼等恶意信息^[26]，这些虚假账号和被盗用的账号统称为异常用户。由于社交网络用户之间本身具有信任关系，其中的恶意信息比传统垃圾邮件等更加危险。如 Twitter 中每天新增 300 万条垃圾信息^[9]，研究发现 Twitter 中垃圾广告链接的点击率比垃圾邮件中链接点击率要高出两个数量级^[18]。

社交网络中的异常用户通过一些不同于正常用户的恶意行为谋取利益，如图 1-3 所示^[19]，这些恶意行为对正常用户的隐私信息、账号安全以及使用体验造成了严重威胁，同时异常用户进行的恶意互粉^[24]、添加好友^[28]、点赞^[12]等行为也严

重危害到在线社交网络的信誉评价体系以及用户之间的信任关系。因此，社交网络中异常用户的检测问题对于社交网站安全、用户隐私保护等具有直接的意义和价值，国内外许多大学和研究机构都在此领域展开了深入研究，如加州大学伯克利分校、卡内基梅隆大学、清华大学、北京大学、微软研究院等，一些重要的研究成果也频频出现在 CCS、WWW、KDD、AAAI 等国际信息安全领域和数据挖掘领域的顶级会议和期刊上。

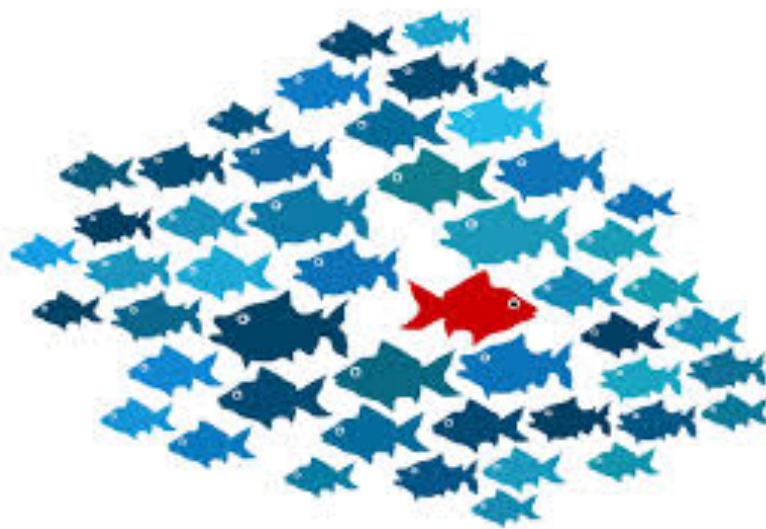


图 1-3：异常用户的行为往往不同于正常用户

1.2 课题的核心问题

总体而言，社交网络中异常用户的检测主要涉及三个核心问题：异常用户的表现、检测方案的设计、检测方案的验证。

1.2.1 异常用户的表现

由于社交网络拥有海量注册用户，这些用户具有形态各异的表现，并且用户的表现是一个动态过程，在不同的阶段具有不同的行为特征，因此在对社交网络中的所有用户进行异常检测之前，首要的问题是如何确定异常用户的表现，从而可以对异常用户有一个合理的定义。

1.2.2 检测方案的设计

在确定异常用户表现的基础之上，面对社交网络中用户纷繁复杂的数据和行为，如何选择合适的特征和算法来设计既满足准确率又满足效率的检测方案，是该领域的核心问题之一。

1.2.3 检测方案的验证

设计的检测方案只有采用真实数据验证后才能证明有效，而社交网络由于涉及商业利益和用户隐私等问题对于数据的获取和使用有严苛的条件，因此如何获取相应的实验数据、对实验结果进行验证也是需要重点关注的问题。

1.3 课题的主要挑战

社交网络海量的用户数、异常用户的多种表现形式以及异常用户特征的动态变化等都为异常用户的检测带来了巨大的挑战。

1.3.1 社交网络巨大的用户数据

社交网络拥有海量的用户数，如 Facebook 在 2016 年已经拥有 18.6 亿活跃用户，用户每天发布的内容以及用户的行为操作更是数不胜数，如 Twitter 每天用户发布的消息达到 5 亿条，而异常用户检测系统需要对每个用户的数据都进行计算，这将花费大量的时间，而异常用户的检测期望能尽早的发现异常用户，降低对正常用户的损害。因此社交网络巨大的用户数据对于异常用户的检测是一个挑战。

1.3.2 异常用户的多种表现形式

社交网络中的异常用户具有多种表现形式，而且不同攻击者创建的异常账号也具有不同的行为模式，同时社交网络拥有海量的用户，这些用户在使用社交网络时本身也表现出不同的行为模式，有经验的攻击者会刻意将异常账号的表现接

近于正常用户，有些甚至正常用户都无法分辨清楚，这使得对于异常用户的检测更加困难。

另外有时需要区分具体的异常用户类型，从而由社交网络服务提供商采取不同的处理方式，比如对于攻击者创建的虚假账号可以直接禁用，而对于被盗用的账号则需要给用户发送安全提醒或者对账号重置密码。因此异常用户的多种表现形式对于异常用户的检测是一个挑战。

1.3.3 异常用户特征的动态变化

异常用户的检测是一个猫鼠游戏，当社交网络根据异常用户的某些特征部署了相应的检测系统之后，攻击者在利益的驱使下总是能够很快找到绕过检测的方式，使异常用户表现出新的特征，检测系统就要重新对特征进行训练，这样社交网络对异常用户的检测往往滞后于攻击者，使得正常用户依然受到异常用户的危害。因此异常用户特征的动态变化对于异常用户的检测是一个挑战。

1.4 课题的研究现状

针对社交网络中异常用户所带来的威胁，学术界和工业界都提出了大量的检测方案。社交网络中异常用户的检测涉及多个领域，如一般数据异常检测、图中异常检测以及垃圾信息检测等。

国内外很多学者对这些方案进行了归纳总结，如 Chandola 等人^[8]对一般性的异常检测方法进行了总结，Akoglu 等人^[5]也是介绍了通用的图中异常检测方法，莫倩等人^[2]介绍了网络中垃圾信息的检测方法。

张玉清等人^[1]根据这些方案所采用核心算法的不同将这些方案分为 4 类，分别为基于行为特征的检测方案、基于内容的检测方案、基于图的检测方案和无监督学习的检测方案。

1.4.1 基于行为特征的检测方案

由于异常用户的主要目的是通过恶意行为如发布广告、钓鱼消息等从中获取利益，而且异常用户往往是通过自动化工具来控制，为了获取利益的最大化，异常用户会提高发布消息的频率或者在短时间内发出大量的好友请求等。因此异常用户与正常用户在某些行为特征方面必然存在一些差异。基于行为特征的检测方案将异常用户检测看为数据挖掘中的分类问题，检测方案的基本流程如下：首先在社交网络中获取数据训练集，然后从数据中抽取相应的行为特征，再利用分类算法对这些特征进行训练形成分类器，最后利用测试样本集对分类器进行测试并判断分类结果。选取的特征主要分为4类，表格1-1列出了常见的特征。

表格 1-1：特征列表

类别	特征
用户个人信息	用户名长度、用户简介长度、账号注册时间、用户名命名规则、被访问次数
用户行为	消息发布时间间隔、评论回复时间、账号注册流程、账号点击顺序、点赞数
好友关系	好友数、粉丝数、好友请求 / 好友数、关注数 / 粉丝数、好友网络聚类系数、二阶好友数、二阶好友消息数
消息内容	消息中 URL 比率、#比率、@比率、消息相似度、消息单词数、消息字符数、评论数、Spam 关键词数、消息来源、消息数、消息转发次数

1.4.2 基于内容的检测方案

异常用户通过发布广告、钓鱼等消息来获取利益，因此在发布的消息内容方面异常用户与正常用户之间存在区别。基于内容的检测方案是利用异常用户所发

布内容与正常用户所发布内容的不同来进行检测，因此检测的重点放在判断用户发布的消息是否为恶意消息。根据不同的消息内容利用对象，将基于内容的检测方案分为以下两类，一类为利用单个账号的内容特征，另一类为利用群体账号的内容特征。

利用单个账号内容特征是根据单个异常账号发布的消息内容如消息中嵌入的 URL 以及发布消息的行为方式与正常用户的区别等来检测异常账号。异常账号在发布的恶意信息中嵌入了指向广告、钓鱼、恶意软件下载等网址的 URL，可以通过判断消息内容中嵌入的 URL 是否恶意来判断发布消息的账号是否异常。通过对单个账号的消息内容特征的变化来检测异常。对于被攻击者劫持的账号，由于其被劫持前后所发布消息的内容和行为有巨大的变化，因此可以通过消息内容特征的变化来检测此类异常账号。

攻击者为了扩大恶意消息的传播范围来获取更多的利益会控制大量的异常账号发布相同或相似的恶意消息，因此也可以利用群体账号的消息内容特征来检测异常账号。例如通过判断消息或消息中包含的 URL 是否相似来检测异常。

1.4.3 基于图的检测方案

社交网络的一个重要特性就是用户之间存在联系，如好友关系、关注、粉丝等，而且也只有两个用户之间存在联系时才能够进行信息交流，因此社交网络中用户之间的关联关系具有图的性质。基于图的检测方案是利用正常用户和异常用户所形成的图中具有不同的结构模式或连接方式，将异常用户检测问题转化为图中异常检测问题，再利用图挖掘的相关算法来区分正常用户与异常用户。

在社交网络中存在众多的图结构，除了显性的好友关系图（如 Facebook 中好友关系组成的图、Twitter 中关注关系组成的图），还存在利用其他关系建立的隐性图结构，如访问关系、分享关系、URL 共享关系等。

1.4.4 无监督学习的检测方案

无监督学习的方法是基于正常用户有相同的特征或者符合一定的模型，通过特征的聚类或者建立模型来检测异常用户。

基于聚类的方案将异常用户检测看为数据挖掘中聚类问题。通过用户的某些特征进行聚类，将正常用户聚为一类，而不在类中的用户即为异常用户；或正常用户聚为一类同时异常用户也聚为一类，通过对类中用户进行抽样验证就能够判断该类内的其他用户是否为异常。因此不需要提前对样本数据进行标识。

基于模型的检测方案的基础是认为正常用户的行为符合某种模型，而异常用户的行为不符合这种模型，因此基于模型的方案的关键是抽取合适的特征对正常用户进行训练，形成相应的模型，然后根据其他用户是否与模型匹配来判断是否为异常用户。

无监督学习的检测方案是目前异常用户检测的新方向。无监督学习方案不需要提前对样本进行标识，因此能够检测到未知的恶意行为。

1.5 本文的工作

根据 1.4.3 节中提到的社交网络用户之间的连接关系具有图的性质，本文希望提出一个基于社交网络的图结构特征的、能够处理大规模网络的异常用户检测算法。

首先，本文基于一个观察提出了一个对于异常用户的合理的定义。在一个社交网络中有些用户节点往往会同时连接多个不同的社区，在进行社区发现时这些

用户节点的存在会严重影响到找到的社区质量，而如果将这些用户节点从社交网络中除去，则会大大提高发现的社区质量，所以本文定义这一类节点所代表的用户为异常用户。

接着，受到近几年新提出的图向量表征算法的启发，本文提出了基于图向量表征的异常用户检测算法。通过图向量表征算法首先将社交网络中的用户节点进行图向量表征，得到每一个用户节点的特征向量。然后，本文基于以下假设，在同一个社区中的用户节点它们的特征向量应该较为接近，不同社区中的用户节点它们的特征向量应该较为疏远，所以对于只属于一个社区的正常用户，他们属于这个社区的概率应该远远大于属于其他社区的概率，然而异常用户由于同时连接多个社区，它们属于每个社区的概率倾向于比较均匀，从而我们可以通过这个明显的区别在社交网络中找出异常用户。

最后，根据本文提出的基于图向量表征的异常用户检测算法，我们使用了真实的社交网络数据集进行实验，并对实验结果进行了分析。

1.6 论文结构简介

论文总共分为六个章节：

第一章：阐述课题的背景和意义、核心问题以及主要挑战，介绍课题的研究现状，最后简单描述本文的工作。

第二章：综述目前已有的异常用户检测算法，详细介绍其中比较具有代表性的三种检测算法。

第三章：介绍本文使用的图论相关基本概念与定义，提出本文对于社交网络中异常用户的定义。

第四章：介绍本文使用的 Network Embedding 算法，提出本文的基于图向量表征的异常用户检测算法，细述算法的具体过程，同时通过每一步的推导论证其合理性。

第五章：介绍本文的实验过程，分析本文的异常用户检测算法应用在真实的社交网络数据集上的实验结果。

第六章：总结本文的工作并分析存在的不足，最后提出未来可以进行改进的方向。

第2章 异常用户检测算法综述

针对社交网络中异常用户所带来的威胁，学术界和工业界都提出了大量的检测算法。本文首先将简单介绍其中比较具有代表性的三种，根据检测算法核心思想的不同，分为基于全局统计规律、基于图的局部结构、基于提取特征的训练三种检测算法。

2.1 基于全局统计规律的检测算法

社交网络是一种典型的无标度网络，网络中节点的度分布符合幂律分布，在该网络中具有少量的高度节点和大量的低度节点。所谓幂律是指节点具有的连线数和这样的节点数目乘积是一个定值，也就是几何平均是定值。幂律分布的通式可写为：

$$y = cx^{-r} \quad (2.1)$$

其中 x 、 y 是正的随机变量， c 、 r 均为大于零的常数。幂律分布表现为一条斜率为幂指数的负数的直线。

图 2-1 为知乎的关注数量分布图，其中横坐标为用户的关注数量的对数，纵坐标为对应的用户数量的对数，这就是一个幂律分布。如图 2-1 所示，红圈标注的点偏离了回归线，即它关注的用户数量远远大于其他一般用户，所以可以认为这就是一个异常用户。

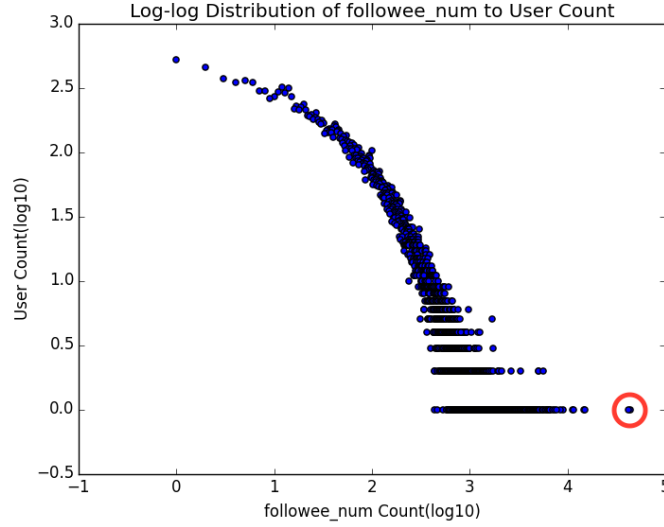


图 2-1：幂律分布

Akoglu 等人提出的 OddBall 算法^[4]是这一类图中异常检测算法的最早的经典著作之一，该算法指出自我中心网络（Egonet）的多个特征的数值之间服从幂律分布。这里，自我中心网络是指由一个节点与其直接邻居所生成的子图。该算法提出一种发现，对于一个自我中心网络，其节点数与边数、边权之和与边数等特征的数值之间均服从幂律分布。因此，该算法提出了一种对于节点异常程度的衡量标准。对于节点 i 的某一个特征对 $f(x, y)$ ，假设其符合表达式为 $y = Cx^\theta$ 的幂律分布，那么节点 i 的异常系数可以表示为：

$$outline(i) = \frac{\max(y_i, Cx_i^\theta)}{\min(y_i, Cx_i^\theta)} * \log(|y_i - Cx_i^\theta| + 1) \quad (2.2)$$

这种衡量标准是以点到拟合直线的距离作为异常程度的参考，也就是利用与其他节点之间特征的数值上的偏离程度来检测出异常用户。

2.2 基于图的局部结构的检测算法

基于全局统计规律的检测算法虽然能够比较简单的通过数值计算找出异常用户，但是只考虑了数值上的特征，并没有考虑图本身的一些结构性性质，所以又有人提出了基于图的局部结构的检测算法。

异常用户的出现往往会导致网络中形成特殊的结构，所以可以通过在网络中寻找这样的结构从而发现异常用户^[22]。例如，Shrivastava等人^[23]提出了社交网络中的一种星状结构可以应用于异常用户的检测。垃圾邮件往往是从单一的一个恶意个体发送至许多的收件人的邮箱中，由于这些收件人是被随机选择的，所以独立于那个恶意个体之后，他们之间很有可能是没有连接的，这样就导致在网络中形成了一种放射形的星状结构，如图2-2的左图所示。而对于一个普通的用户节点，如果与它连接的两个用户节点也相互连接，这样三条边就会形成一个三角形的环，如图2-2的右图所示。基于这个观察，该算法提出可以通过计算用户节点所在三角形的个数与边数的比例判断其是否为异常用户。一般来说，较少的三角形比例往往意味着这个用户是异常用户的可能性较大。

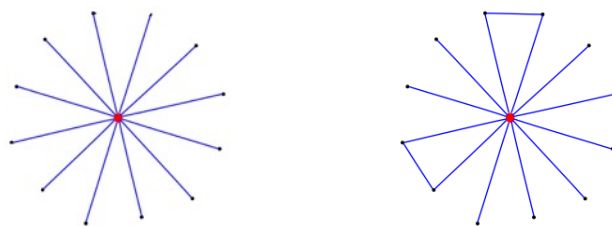


图 2-2：星状结构

2.3 基于提取特征的训练的检测算法

基于图的局部结构的检测算法可以比较直观的通过结构特征找出异常用户，但是这种方法相对复杂度较大，并不是很适合于大规模的社交网络，所以基于提取特征的训练的检测算法被提出。

这一类算法一般通过人为选择提取的社交网络特征进行有监督或无监督的学习，生成正常用户或异常用户的特征模型，然后再进行用户与模型的匹配从而检测出异常用户。例如，Bhat 等人^[6]基于社区对节点的特征进行提取，包括节点的出入度之比、是否为社区边缘、跨社区边数、跨社区出入度之比等特征，然后

使用了决策树、朴素贝叶斯、K 最近邻等分类算法进行模型的训练，最后找出社交网络中的异常用户。

这一类算法虽然也能找出与其他大部分用户在选取的特征上区别较大的异常用户，但是这样人为挑选的特征往往并不充分，不足以完全表达社交网络中的一些潜在信息，从而导致模型的不准确。另外，对于有监督学习的方法来说，数据集必须本身带有是否为异常用户的标签，然而这样的数据集往往很难获取，所以也不能很好的解决异常用户的检测问题。

第3章 图论相关知识与异常用户定义

3.1 图论的相关知识

由于本文涉及到图论的相关知识，所以为了方便，首先对图论的一些基本概念和定义做出介绍。

3.1.1 图

对于一个图 $G(V, E)$, $V = \{v_1, v_2, \dots, v_n\}$ 表示图中所有节点的集合, $n = |V|$ 为节点的数目, E 表示图中所有边的集合, $(v_i, v_j) \in E$ 表示图中一条连接节点 v_i 和 v_j 的边。

3.1.2 度

在图 $G(V, E)$ 中, 节点 v 的邻居表示为 $neighbors(v) = \{u | (u, v) \in E\}$, 从而节点 v 的度表示为 $degree(v) = |neighbors(v)|$, 即节点 v 所连接的边的数目。

3.1.3 社区

社交网络具有模块结构的特性, 在社交网络中不同用户之间通过互相关注建立联系, 多个紧密连接的用户之间会形成一个个社区。同一社区内的节点之间连接比较紧密, 而社区与社区之间的连接却是比较稀疏的。在同一个社区中的用户往往具有某些相似的特征, 这一特点可以应用于信息的推送、商品的推荐等个性化服务, 对在社交网络中寻找用户群体这一问题的研究称为社区发现。

3.2 异常用户的定义

对于社交网络中的异常用户有诸多不同的定义, 在不同的情况下的定义并没有一个明确统一的标准, 所以本文首先提出一个对于异常用户的合理的定义, 作为检测算法的立足基础, 定义基于下面一种观察。

在现实生活的社交网络中，经常会出现一个用户同时连接了属于多个社区的不同用户的情况。例如，一些营销账号为了扩大利益会主动去添加非常多的陌生人为好友，然而他的这些好友之间却并不是或不能形成一个真正的社区，如图 3-1 所示。这就使得对社交网络进行社区发现时会得到质量不高的结果，但是如果将这些用户从社交网络中去除，则会使得发现的社区质量明显提高，所以我们定义这样连接了多个不同社区的用户为异常用户。



图 3-1：同时连接了多个社区的异常用户

为了更加直观的描述这种观察，我们利用一张相对简单的图进行进一步的解释。如图 3-2 所示，四种不同的颜色清晰的显示了网络中四个不同的社区，中间红色节点所示的用户同时连接了这四个不同的社区，而不是仅仅连接了其中一个。这样就使得这四个社区之间的界限没有那么明确，从而增加了社区发现的难度。所以这个用户被我们判断为异常用户，本文检测算法的目标就是在社交网络中找出类似这样的用户，作为异常用户将其从中除去。

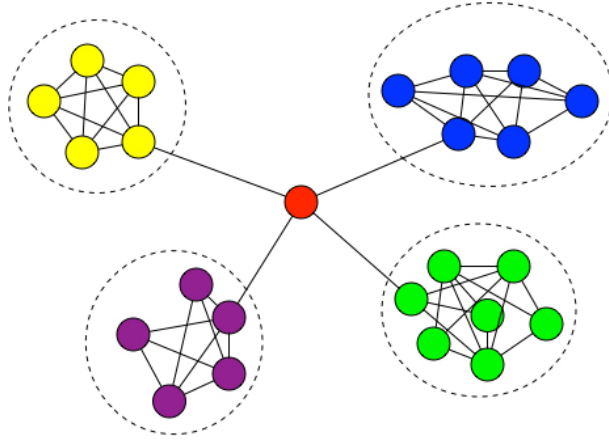


图 3-2: 异常用户（红色节点）的简单例子

其次，我们提出一个假设，假设一个用户属于某一个社区，那么他关于这个社区的隶属度应该相对比较大，而关于其他社区的隶属度应该比较小或者接近于零。这样对于一个异常用户来说，由于他同时连接多个不同的社区，所以与多个社区都有一定的相关性，那么他关于所连接的每个社区的隶属度应该倾向于相对比较均匀。

基于上述观察和假设，下面我们提出对于这种异常用户的数学定义。

定义: 异常用户。 在一个社区数量为 k 的社交网络图中，对于 $\forall v_i \in V$ ，令 $\mathbf{y}_i = (y_i^{(1)}, y_i^{(2)}, \dots, y_i^{(k)})$ 为用户节点 $v_i \in V$ 的隶属度向量， $y_i^{(j)}$ 表示用户节点 v_i 关于第 j 个社区的隶属度，其中 $0 \leq y_i^{(j)} \leq 1$ 且 $\sum_{j=1}^k y_i^{(j)} = 1$ 。计算向量 \mathbf{y}_i 中所有非零项的平均值 Avg 、最大值 Max ，以及大于等于最大值的项数 Cnt ，如果 $\frac{Avg(\mathbf{y}_i)}{Max(\mathbf{y}_i)} \cdot Cnt \geq thre$ ，其中 $thre$ 为人为设定的阈值（threshold），那么 v_i 所代表的用户为异常用户。

为了便于理解，我们再次使用图 3-2 进行进一步的解释。假设该社交网络中只有这四个社区，即 $k = 4$ ，那么在最理想的情况下，四个社区中的用户节点的隶属度向量应该分别为 $(1,0,0,0)$ 、 $(0,1,0,0)$ 、 $(0,0,1,0)$ 、 $(0,0,0,1)$ ，而中间的红色节

点的隶属度向量应该为(0.25,0.25,0.25,0.25)，从而通过定义中的不等式可以将这个异常节点区分出来。

对于社区数量较多的社交网络也是如此。正常用户节点的隶属度向量中应该只有一项较大且接近于 1，其他项较小且接近于 0；相反，异常用户节点虽然不一定会与每个社区都有连接，但是其隶属度向量中会出现多个较大的非零项，且这些非零项的值倾向于相等。所以，通过这个比较明显的区别就可以将异常用户从社交网络中挑选出来。

第4章 基于图向量表征的异常用户检测算法

现有的异常用户检测算法找到异常用户后难以验证其有效性，本身带有标签的社交网络用户数据难以获取。因此，本文通过间接的方法来实现社交网络中异常用户的检测，主要基于用户之间的图结构特征，提出基于图向量表征的异常用户检测算法，将图中异常检测问题转化为一般的数据异常检测问题，通过对节点隶属度向量的学习过程，最终找出存在于社交网络中的异常用户。

为了能够较好的自动提取网络特征，本文使用了近几年新提出的用于处理大规模网络的 Network Embedding 算法，根据社交网络中用户节点之间的连接关系得到每一个用户节点的图表征向量，作为检测算法的输入。所以，这里将首先对 Network Embedding 算法进行介绍，然后再详细的介绍本文提出的基于图向量表征的异常用户检测算法的具体内容。

4.1 Network Embedding

Network Embedding 是近几年新提出的一种很重要的将网络的节点用低维向量表征的学习方法，其目标是能够很好的提取和保留网络的结构^[27]。目前广泛应用于相似节点计算、链路预测、社团划分、图可视化等领域，其同义词有 Graph Embedding 和 Graph Representation。

Word Embedding，即词向量表征法，是比较流行的一种将每一个单词转化为一个向量的学习方法，那么 Network Embedding 则是图向量表征的方法。下面给出 Network Embedding 的定义。

定义: Network Embedding. 给定网络 $G = (V, E)$ ， V 是网络节点集合， E 是边的集合，寻找一个映射函数：

$$f_G: V \rightarrow R^d, d \ll |V| \quad (4.1)$$

将网络的节点映射到低维向量空间 R^d ，转换后的低维向量尽可能保留网络的特性。

给定一个网络，Network Embedding 的目标是寻找一个映射函数，将图数据映射到一个低维的潜在空间，每一个节点用一个低维向量来表示，从而可以直接实现对于网络的计算，网络中的信息挖掘，例如信息检索、分类分析、聚类分析等，就可以直接在这个低维空间中进行。

Network Embedding 将网络的节点映射到低维的向量空间，映射后的低维向量表征应该依然能够尽可能的保留网络的特性。比如网络中两个节点是相邻或者连通的，那么表征后的向量也应该尽可能的相似，而不相邻的或者不属于同一个连通分支的两个节点，它们的表征向量之间的相似度也应该较低。

目前常用的 Network Embedding 算法主要有以下三种，按照提出的时间顺序分别为 DeepWalk^[21]、LINE^[25]、node2vec^[16]。

4.1.1 DeepWalk

DeepWalk 算法第一次将语言模型中著名的 word2vec 模型^[20]引入到图向量表征中。word2vec 模型能够把单词映射到 k 维的向量空间，使得 k 维向量空间的相似度等于文本语义的相似度。DeepWalk 算法就是引用了 word2vec 模型中这种词表征前与词表征后相似性一致的思想。

DeepWalk 算法主要包含两部分，随机游走（Random Walk）以及一个更新过程（Skip Gram），其伪代码如图 4-1 所示。

Algorithm 1 DEEPWALK(G, w, d, γ, t)

Input: graph $G(V, E)$ window size w embedding size d walks per vertex γ walk length t **Output:** matrix of vertex representations $\Phi \in \mathbb{R}^{|V| \times d}$ 1: Initialization: Sample Φ from $\mathcal{U}^{|V| \times d}$ 2: Build a binary Tree T from V 3: **for** $i = 0$ to γ **do**4: $\mathcal{O} = \text{Shuffle}(V)$ 5: **for each** $v_i \in \mathcal{O}$ **do**6: $\mathcal{W}_{v_i} = \text{RandomWalk}(G, v_i, t)$ 7: $\text{SkipGram}(\Phi, \mathcal{W}_{v_i}, w)$ 8: **end for**9: **end for**

图 4-1: DeepWalk 算法伪代码

在一个语言建模算法中,所需的输入是一个词库和一个语料库,DeepWalk 算法将图中的所有节点作为自己的词库,将一系列短距离随机游走生成的节点序列作为自己的语料库。DeepWalk 算法每次任意选取一个节点作为起始节点,从这个节点开始,随机选择一个它的相邻节点作为序列中的下一个节点,再从这个被选中的节点开始继续这个随机过程,直到达到最大路径长度生成节点序列,也就是“句子”或上下文环境,从而获得了网络的结构信息。

Skip Gram 是 word2vec 中的语言模型,其思想是给定句子中的某个词,最大化一个句子中在其左右一定窗口范围中出现的词的出现概率^[1]。在 DeepWalk 算法中则是,给定某个节点 v_i ,最大化一个节点序列中在其左右一定窗口范围($i - w, i + w$)中周围的节点的出现概率,优化目标函数可以表示为:

$$\min \Phi = -\log \Pr(\{v_{i-w}, \dots, v_{i-1}, v_{i+1}, \dots, v_{i+w}\} | \Phi(v_i)) \quad (4.2)$$

其中 Φ 是将节点映射到向量空间的表示矩阵。算法伪代码如图 4-2 所示。逐次迭代更新路径中每一个节点的表征向量,实现对上述优化目标函数的求解,而

这个向量表征是可以代表网络节点的相互关系的，最后在图结构中相互连接或者连接关系邻近的节点在低维向量空间中的的向量也会更加靠近。

Algorithm 2 SkipGram($\Phi, \mathcal{W}_{v_i}, w$)

```

1: for each  $v_j \in \mathcal{W}_{v_i}$  do
2:   for each  $u_k \in \mathcal{W}_{v_i}[j - w : j + w]$  do
3:      $J(\Phi) = -\log \Pr(u_k | \Phi(v_j))$ 
4:      $\Phi = \Phi - \alpha * \frac{\partial J}{\partial \Phi}$ 
5:   end for
6: end for

```

图 4-2: Skip Gram 算法伪代码

4.1.2 LINE

LINE 算法是 Large-scale Information Network Embedding 的缩写。该算法设计了两个目标函数，在图向量表征中保留网络的 1 阶相似度（First-order Proximity）和 2 阶相似度（Second-order Proximity）。下面分别给出 1 阶相似度和 2 阶相似度的定义。

定义: First-order Proximity. 对于一条边 (u, v) ，其权重 w_{uv} 就是节点 u 和 v 之间的 1 阶相似度。如果节点 u 和 v 之间没有边，那么它们的 1 阶相似度为 0。

定义: Second-order Proximity. 令 $p_u = (w_{u,1}, \dots, w_{u,|V|})$ 为节点 u 与其他所有节点之间的 1 阶相似度，那么节点 u 和 v 之间的 2 阶相似度则为 p_u 和 p_v 之间的相似度。如果没有节点同时与节点 u 和 v 相连，那么它们之间的 2 阶相似度为 0。

一个直观的解释是，在一个社交网络中，A 与 B 是好友，那么如果对这两个节点进行图向量表征，它们表征后的向量之间的距离应该是比较小的，这就是网络的 1 阶相似关系。而如果 C 与 D 并不是好友，但他们有很多的共同好友，那么对这两个节点图向量表征后，也应该能够保留这一层的关系，LINE 算法中称之为 2 阶相似关系。

对于 1 阶相似度，定义每条无向边 (i, j) 的两个节点 v_i 和 v_j 的联合概率为：

$$p_1(v_i, v_j) = \frac{1}{1 + \exp(-u_i^T \cdot u_j)} \quad (4.3)$$

其中 $u_i \in R^d$ 是节点 v_i 的低维向量表示，所以为保留 1 阶相似度则需要最小化目标函数：

$$O_1 = - \sum_{(i,j) \in E} w_{ij} \log p_1(v_i, v_j) \quad (4.4)$$

对于 2 阶相似度，引入 u_i' 表示节点 v_i 的上下文环境，定义每条有向边 (i, j) 的节点 v_j 对 v_i 的条件概率为：

$$p_2(v_j | v_i) = \frac{\exp(u_j'^T \cdot u_i)}{\sum_{k=1}^{|V|} \exp(u_k'^T \cdot u_i)} \quad (4.5)$$

其中 $|V|$ 是节点或上下文环境的数量，所以为保留 2 阶相似度则需要最小化目标函数：

$$O_2 = - \sum_{(i,j) \in E} w_{ij} \log p_2(v_j | v_i) \quad (4.6)$$

通过分别最小化上述两个目标函数，可以分别得到保留网络的 1 阶相似度和 2 阶相似度的图表征向量，为了同时保留 1 阶相似度和 2 阶相似度，LINE 算法提出了一个简单有效的方法则是将每个节点两次训练后得到的向量串联起来，从而得到最终的图表征向量。

4.1.3 node2vec

node2vec 算法将网络中的特征学习规定为最大可能性优化问题。令 $f: V \rightarrow R^d$ 为由节点到特征向量的映射函数，其中 d 为特征向量的维度。对于每一个节点 $u \in V$ ，定义 $N_S(u) \subset V$ 为节点 u 通过邻居采样策略 S 生成的网络邻居，从而需要优化的目标函数为：

$$\max f = \sum_{u \in V} \log \Pr(N_S(u) | f(u)) \quad (4.7)$$

一般来说，生成网络邻居的采样策略主要有宽度优先采样（BFS）和深度优先采样（DFS）两种。如图 4-3 所示，假设网络邻居集合的大小为 3，那么 BFS 会采样 s_1 、 s_2 、 s_3 三个节点，而 DFS 会采样 s_4 、 s_5 、 s_6 三个节点。

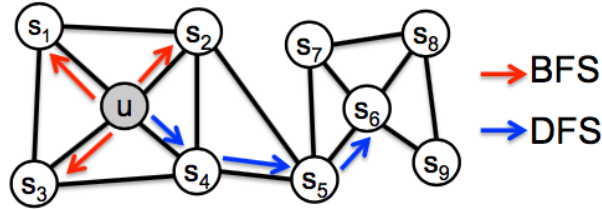


图 4-3：从节点 u 出发的 BFS 与 DFS 两种搜索策略

node2vec 算法基于两个假设，分别为同质性假设和结构等价假设。同质性假设指的是在同一社区的节点图向量表征应该尽可能相似，如图 4-3 中的节点 u 和 s_1 ；而结构等价假设指的是网络中充当相同结构角色的节点图向量表征应该尽可能相似，如图 4-3 中的节点 u 和 s_6 。

不同的网络搜索模式，会得到不同的节点序列，从而会生成不同的图向量表征。利用 BFS 生成的节点序列一般多集中于网络中的某个局部结构内，学习出的节点表达倾向于表示节点之间的结构等价性；利用 DFS 生成的节点序列一般能较好的遍历整个网络，学习出的节点表达可以显示出具有同质性的社区结构。

node2vec 算法提出了一种可调节的 2 阶随机游走方式，引入了用以调节、平衡这两种搜索采样方式的 Search bias 函数。如图 4-4 所示， t 为上一次遍历的节点， v 为当前所在的节点， x 为待遍历的节点。

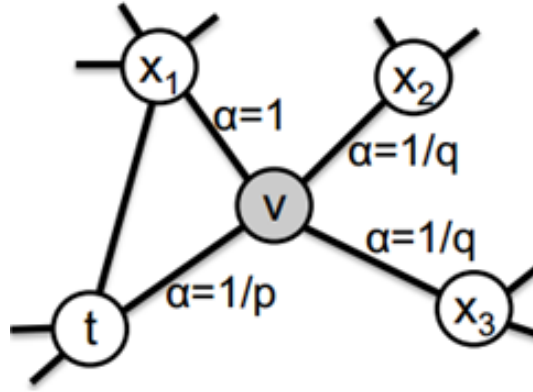


图 4-4: node2vec 随机游走过程

Search bias 是一个阶梯函数:

$$\alpha_{pq}(t, x) = \begin{cases} \frac{1}{p} & \text{if } d_{tx} = 0 \\ 1 & \text{if } d_{tx} = 1 \\ \frac{1}{q} & \text{if } d_{tx} = 2 \end{cases} \quad (4.8)$$

d_{tx} 表示节点 t 和 x 之间的最短路径长度, w_{vx} 为边 (v, x) 的权重, 令 $\pi_{vx} = \alpha_{pq}(t, x) \cdot w_{vx}$ 为节点 v 与 x 之间的状态转移概率, 从而可以使用 p 和 q 两个参数决定随机游走的下一个节点。调节这两个值, 就可以得到不同的图向量表征。

Algorithm 1 The *node2vec* algorithm.

```

LearnFeatures (Graph  $G = (V, E, W)$ , Dimensions  $d$ , Walks per
node  $r$ , Walk length  $l$ , Context size  $k$ , Return  $p$ , In-out  $q$ )
   $\pi = \text{PreprocessModifiedWeights}(G, p, q)$ 
   $G' = (V, E, \pi)$ 
  Initialize walks to Empty
  for  $iter = 1$  to  $r$  do
    for all nodes  $u \in V$  do
       $walk = \text{node2vecWalk}(G', u, l)$ 
      Append  $walk$  to walks
   $f = \text{StochasticGradientDescent}(k, d, \text{walks})$ 
  return  $f$ 

node2vecWalk (Graph  $G' = (V, E, \pi)$ , Start node  $u$ , Length  $l$ )
  Initialize  $walk$  to  $[u]$ 
  for  $walk\_iter = 1$  to  $l$  do
     $curr = walk[-1]$ 
     $V_{curr} = \text{GetNeighbors}(curr, G')$ 
     $s = \text{AliasSample}(V_{curr}, \pi)$ 
    Append  $s$  to  $walk$ 
  return  $walk$ 

```

图 4-5: node2vec 算法伪代码

node2vec 算法的伪代码如图 4-5 所示。

4.2 基于图向量表征的异常用户检测算法

上文介绍了对于异常用户的定义和 Network Embedding 算法，下面正式介绍本文提出的基于图向量表征算法的异常用户检测算法的具体内容。检测算法主要分为用户节点特征提取、目标函数设计与求解、隶属度向量学习以及异常用户判断四个过程，下面将逐一进行介绍。

4.2.1 特征的提取

对于一个给定的社交网络，我们第一步需要获得图中所有用户节点的特征向量。这里我们选择使用上文提到的 node2vec 算法，这是由于 node2vec 算法总结了前两种算法的思想，通过对其随机游走方式的两个参数进行调节，可以得到适合本文算法的关注社区结构的图向量表征。

node2vec 算法通过读取数据集中用户节点的邻接表，提取每一个节点的图结构特征，从而得到相应的图表征向量。之后，再将我们通过邻接表计算的节点的度（degree）作为补充特征，最后得到所有节点的特征向量。为了方便，我们令节点的特征向量的维度 d ，对于每一个节点 v_i ，我们在这一步得到的其特征向量用 $\mathbf{x}_i = (x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(d)})$ 表示。

4.2.2 目标函数的设计

根据上文 3.2 节中对异常用户的定义，对于每一个节点 v_i ，我们希望得到其关于 k 个社区的隶属度向量 $\mathbf{y}_i = (y_i^{(1)}, y_i^{(2)}, \dots, y_i^{(k)})$ ，其中 $y_i^{(j)}$ 表示节点 v_i 关于第 j 个社区的隶属度，社区数量 k 的值人为设定。为了使学习出的隶属度向量尽可能

好的将社区中的正常用户节点与同时连接多个社区的异常用户节点区分开来，我们设计了如下的目标函数 O ：

$$\min O = \sum_{i=1}^n \sum_{j=1}^k y_i^{(j)} \cdot \left\| \mathbf{x}_i - \mathbf{m}_j \right\|_2^2 + \alpha \sum_{i=1}^n \left\| \mathbf{y}_i \right\|_2^2 \quad (4.9)$$

约束条件为 $0 \leq y_i^{(j)} \leq 1$ 且 $\sum_{j=1}^k y_i^{(j)} = 1$ 。其中， \mathbf{m}_j 表示第 j 个社区质心的特征向量， $\left\| \mathbf{x}_i - \mathbf{m}_j \right\|_2^2$ 与 $\left\| \mathbf{y}_i \right\|_2^2$ 表示向量 $\mathbf{x}_i - \mathbf{m}_j$ 与 \mathbf{y}_i 的 2-范数（Euclid 范数）的平方，即向量每一项元素的平方之和， α 为人为设定的参数，下文中会详细介绍其作用。目标函数 O 中包含 $\sum_{i=1}^n \sum_{j=1}^k y_i^{(j)} \cdot \left\| \mathbf{x}_i - \mathbf{m}_j \right\|_2^2$ 与 $\alpha \sum_{i=1}^n \left\| \mathbf{y}_i \right\|_2^2$ 两项，我们的优化目标是将这两项之和最小化。下面分别介绍这两项表达式的意义。

首先对于前一项，其值等于所有 n 个节点分别关于所有 k 个社区的隶属度与该节点向量与该社区质心向量的距离平方的乘积的总和。如果将这一项最小化，那么显然，每个节点应该尽可能被判定为正常节点，其隶属度向量应该趋向于 $(1,0,0, \dots, 0)$ 形式，只有一维的值等于或接近于 1，其他维的值等于或接近于 0。

而对于后一项，不考虑参数 α 的前提下，其值等于所有 n 个节点的隶属度向量的模长的平方之和。如果将这一项最小化，那么显然，每个节点应该尽可能被判定为异常节点，其隶属度向量应该趋向于 $(\frac{1}{k}, \frac{1}{k}, \frac{1}{k}, \dots, \frac{1}{k})$ 形式，对于所有值非零的维，其值分布比较均匀。

这样，我们对目标函数的最小化过程其实就是前后两项之间的一种博弈，所以我们通过设置参数 α 进行两者之间的权衡（trade-off），从而调节两者在优化过程中起到的作用大小。例如，当参数 α 的值较小时，在目标函数中起主导作用的是第一项，这时学习出的每一个节点的隶属度向量趋向于 $(1,0,0, \dots, 0)$ 形式；而当参数 α 的值较大时，在目标函数中起主导作用的是第二项，这时学习出的每一

个节点的隶属度向量趋向于 $(\frac{1}{k}, \frac{1}{k}, \frac{1}{k}, \dots, \frac{1}{k})$ 形式。所以，参数 α 的存在较好的避免了学习出的结果出现上述两种极端情况的可能。

4.2.3 目标函数的求解

为了学习出尽可能好的隶属度向量，下面我们需要对目标函数进行求解，从而得到每次更新隶属度向量的变换公式。显然，目标函数 O 是一个包含 y_i 和 m_j 两个变量的二元凸函数，我们采用每次固定一个变量优化另外一个变量这种交替的更新方式求出其极小值。

(1) 固定 y_i ，优化 m_j 。

对于这样的二元函数，我们采用令偏导数为零的方式求出其极值点。令 $\frac{\partial O}{\partial m_j} = 0$ ，可得

$$\frac{\partial O}{\partial m_j} = 2 \sum_{i=1}^n y_i^{(j)} \cdot (x_i - m_j) = 0 \quad (4.10)$$

从而可得

$$\sum_{i=1}^n y_i^{(j)} \cdot x_i = \sum_{i=1}^n y_i^{(j)} \cdot m_j \quad (4.11)$$

最后得到

$$m_j = \frac{\sum_{i=1}^n y_i^{(j)} \cdot x_i}{\sum_{i=1}^n y_i^{(j)}} \quad (4.12)$$

这样，我们得到了等式(4.12)为变量 m_j 的更新公式。

(2) 固定 m_j ，优化 y_i 。

同理，我们依旧采用令偏导数为零的方式求极值点。但是需要注意的是，因为变量 y_i 具有 $0 \leq y_i^{(j)} \leq 1$ 和 $\sum_{j=1}^k y_i^{(j)} = 1$ 的约束条件，所以我们需要通过拉格朗日乘数法将这个约束条件加入到目标函数中，将其变为无约束最优化问题。

拉格朗日乘数法是一种寻找变量由一个或多个条件所限制的多元函数的极值的方法，通过引入拉格朗日乘数这个新的标量未知数，将一个有 n 个变量与 k 个约束条件的最优化问题转换为一个有 $n + k$ 个变量的方程组的极值问题，其变量不受任何约束。

所以，我们将目标函数 O 修改为

$$\min O = \sum_{i=1}^n \sum_{j=1}^k y_i^{(j)} \cdot \|x_i - m_j\|_2^2 + \alpha \sum_{i=1}^n \|y_i\|_2^2 + \lambda \left(\sum_{j=1}^k y_i^{(j)} - 1 \right) \quad (4.13)$$

令 $\frac{\partial O}{\partial y_i^{(j)}} = 0$ ，可得

$$\frac{\partial O}{\partial y_i^{(j)}} = \|x_i - m_j\|_2^2 + 2\alpha y_i^{(j)} + \lambda = 0 \quad (4.14)$$

解得

$$y_i^{(j)} = -\frac{\|x_i - m_j\|_2^2 + \lambda}{2\alpha} \quad (4.15)$$

令 $\frac{\partial O}{\partial \lambda} = 0$ ，可得

$$\sum_{j=1}^k y_i^{(j)} - 1 = 0 \quad (4.16)$$

将等式 (4.15) 代入等式 (4.16) 替换 $y_i^{(j)}$ 解得

$$\lambda = -\frac{2\alpha + \sum_{j=1}^k \|x_i - m_j\|_2^2}{k} \quad (4.17)$$

再将等式 (4.17) 代入等式 (4.15) 替换 λ 解得

$$y_i^{(j)} = \frac{2\alpha + \sum_{j=1}^k \|x_i - m_j\|_2^2 - k \|x_i - m_j\|_2^2}{2k\alpha} \quad (4.18)$$

这样，我们得到了等式 (4.18) 为变量 $y_i^{(j)}$ 的更新公式。

至此，我们完成了对于目标函数 O 的求解过程，并分别得到了变量 m_j 和 y_i 的更新公式，即等式 (4.12) 和等式 (4.18)。

4.2.4 隶属度向量的学习

在推导出变量 \mathbf{m}_j 和 \mathbf{y}_i 的更新公式之后，我们就可以进行目标函数 O 的优化过程，通过迭代更新变量 \mathbf{m}_j 和 \mathbf{y}_i 的值，求出最终的隶属度向量。

Algorithm Probability Learning[↵]

Input: feature vectors $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ [↵]
number of clusters k [↵]
parameter α [↵]

Output: $y_i^{(j)}, 1 \leq i \leq n, 1 \leq j \leq k$ (possibility of node i belonging to cluster j)[↵]

1: **Initialization:** $y_i^{(j)}, \mathbf{m}_j$ (by K-Means Algorithm)[↵]
2: $t = 0$ [↵]
3: **Repeat**[↵]
4: **Update** \mathbf{m}_j (by equation (4.12))[↵]
5: **Update** $y_i^{(j)}$ (by equation (4.18))[↵]
6: $t = t + 1$ [↵]
7: **Until** convergence or $t > t_{max}$ [↵]

图 4-6：隶属度向量学习算法伪代码

算法的伪代码如图 4-6 所示，其中，输入包括节点特征向量 X 、社区数量 k 以及参数 α ，输出为每一个节点 v_i 关于每一个社区 j 的隶属度 $y_i^{(j)}$ 。

算法的第一步是对变量 \mathbf{m}_j 和 \mathbf{y}_i 的初始化。对于优化迭代问题，一个好的初始化可以减少迭代次数，有利于结果收敛达到全局最优。所以，这里我们使用聚类分析中经典的 K 均值（K-Means）算法。下面首先简单介绍聚类分析的概念以及 K 均值算法的原理。

聚类分析是数据挖掘中的重要任务之一，是一个将数据集中在某些方面相似的数据成员进行分类组织的无监督学习过程。聚类分析将数据分类到不同的类或者簇中，同一个簇中的对象有很大的相似性，而不同簇之间的对象有很大的相异性。与有监督学习中的分类不同，无监督学习不依赖预先定义的类或带类标记的训练实例，需要由聚类学习算法自动确定标记，而分类学习的实例或数据对象有类别标记。

本文使用的 K 均值算法是聚类分析中的经典算法。该算法是典型的基于距离的聚类算法，采用距离作为相似性的评价指标，即认为两个对象的距离越近，其相似度就越大。该算法首先随机选取 K 个对象作为初始的聚类中心，初始的代表一个个簇。然后计算剩余每个对象与各个簇中心之间的距离，把每个对象分配给距离它最近的簇。一旦全部对象都被分配了，每个聚类的聚类中心会根据聚类中现有的对象被重新计算。这个过程将不断重复迭代直到没有对象被重新分配给不同的聚类。

利用 K 均值算法根据节点的特征向量将所有节点聚类为 k 类后，就可以得到每个社区质心的初始特征向量。而且因为这时每一个节点都被划分到某一个社区中，所以我们可以得到每一个节点的初始隶属度向量，在这个初始隶属度向量中，只有代表其归属于的社区的维度值为 1，而其他维度值均为 0。

然后，我们就可以根据 4.2.3 中的两个更新公式对变量 m_j 和 y_i 进行交替迭代更新，直到结果收敛或迭代次数达到规定的最大值。最后我们得到了使目标函数 O 达到最小值时的每一个节点的隶属度向量。

4.2.5 异常用户的判断

在学习出所有节点最终的隶属度向量之后，就可以进行异常用户检测算法的最后一步，根据 3.2 节中异常用户的定义对每一个节点进行判断，找出存在在社交网络中的异常用户。

我们定义异常系数 $AScore$ 作为每一个节点 v_i 是否为异常用户的判断标准，其公式为：

$$AScore = \frac{Avg(y_i)}{Max(y_i)} \cdot Cnt(y_i^{(j)} \geq Avg(y_i)) \quad (4.19)$$

根据每一个节点 v_i 的隶属度向量 \mathbf{y}_i 计算, 如果 $AScore \geq thre$, 其中 $thre$ 为人
为设定的阈值, 那么节点 v_i 所代表的用户为异常用户, 该用户将会被输出并从社
交网络数据集中删除。

至此, 社交网络中的所有异常用户被找出, 基于图向量表征的异常用户检测
算法结束。

第5章 实验过程与结果分析

5.1 数据集的获取

检测方案的实现与验证都需要大量的真实数据，对于个人使用来说，数据的获取方式主要有以下两种：

5.1.1 爬虫获取

社交网络都提供了相应的 API，能够直接利用爬虫程序调用 API 获取账号信息，但是社交网络对 API 的使用有一定的限制。也可利用网络爬虫直接获取，但是这种方式只能获取账号的公开数据。

5.1.2 公开数据集

一些学者公开了自己所获取的数据集，有些机构汇总了相关数据，如斯坦福大学的 SNAP²等对社交网络相关的公开数据集进行了总结，因此可以利用公开数据集进行实验。

通过爬虫获取数据是最常见的数据获取方式，而且能够根据实验需求获取指定数据，但是需要耗费一定的人力成本来编写相应的爬虫以及一定的机器时间来爬取数据。利用公开的数据集可以节约时间成本，而且能够在相同数据集上与其他工作进行对比，但是公开的数据集与实验所需的数据内容不一定完全符合，会对实验结果有一定的影响。

5.2 数据集的描述

本次实验使用的是斯坦福大学的 SNAP 上公开的安然（Enron）公司邮件数据集³。安然公司是世界上最大的综合性天然气和电力公司之一，在北美地区是

² <http://snap.stanford.edu/data>

³ <http://snap.stanford.edu/data/email-Enron.html>

头号天然气和电力批发销售商。本数据集包含 36692 个用户节点以及 183831 条无向边，每一个节点代表一位公司职员，而每一条无向边表示两端节点所代表的公司职员之间存在邮件往来。

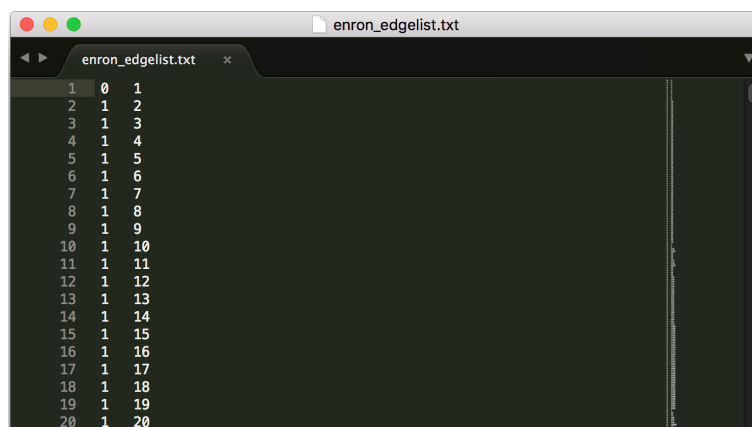


图 5-1：安然公司邮件数据集格式

数据集文件为 `enron_edgelist.txt`，其格式如图 5-1 所示。例如，第一行表示节点 0 与节点 1 之间存在一条无向边。

5.3 实验过程

本文的实验是在处理器为 3.3GHz Intel Core i7、内存大小为 16GB、系统版本为 macOS 10.12.4 的 PC 上进行的。编程语言主要为 Python 和 C++。下面详细介绍实验的具体过程。

5.3.1 生成节点特征向量

上文 4.1 节中介绍了三种常用的 Network Embedding 算法，三种算法的作者均已将自己的算法开源，并且提供了详细的安装及使用说明文档⁴⁵⁶。对于本次实验中的数据集，使用的是 node2vec 算法生成 16 维的图向量表征。

使用 node2vec 算法生成图向量表征的命令如下：

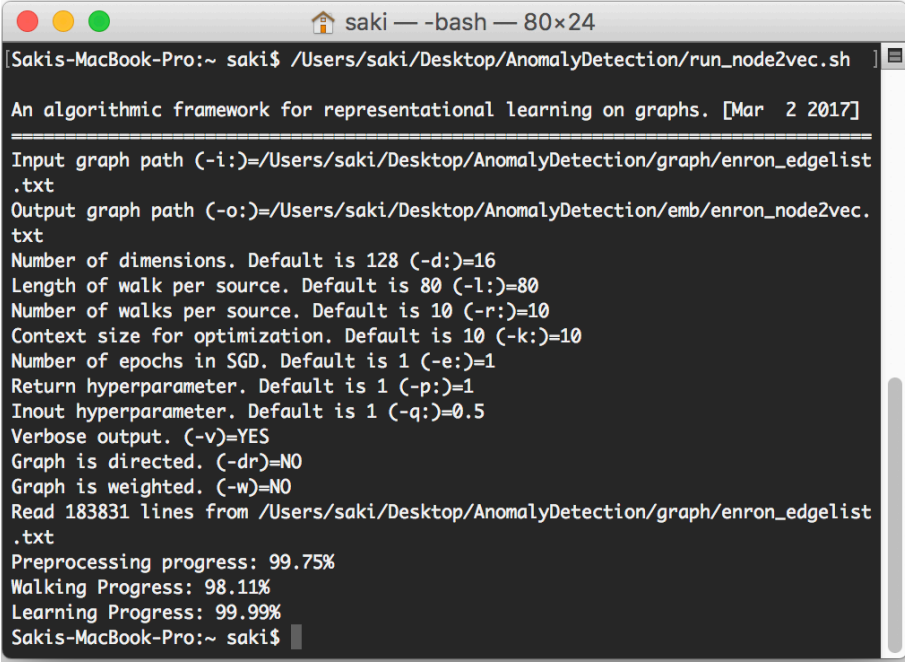
⁴ <https://github.com/phanein/deepwalk>

⁵ <https://github.com/tangjianpku/LINE>

⁶ <https://github.com/snap-stanford/snap/tree/master/examples/node2vec>

`./node2vec -i: enron_edgelist.txt -o: enron_node2vec.txt -d: 16 -p: 1 -q: 0.5 -v`

在命令行中执行以上命令输出如图 5-2 所示：



```
[Sakis-MacBook-Pro:~ saki$ /Users/saki/Desktop/AnomalyDetection/run_node2vec.sh ]
An algorithmic framework for representational learning on graphs. [Mar 2 2017]
=====
Input graph path (-i:)=/Users/saki/Desktop/AnomalyDetection/graph/enron_edgelist.txt
Output graph path (-o:)=/Users/saki/Desktop/AnomalyDetection/emb/enron_node2vec.txt
Number of dimensions. Default is 128 (-d:)=16
Length of walk per source. Default is 80 (-l:)=80
Number of walks per source. Default is 10 (-r:)=10
Context size for optimization. Default is 10 (-k:)=10
Number of epochs in SGD. Default is 1 (-e:)=1
Return hyperparameter. Default is 1 (-p:)=1
Inout hyperparameter. Default is 1 (-q:)=0.5
Verbose output. (-v)=YES
Graph is directed. (-dr)=NO
Graph is weighted. (-w)=NO
Read 183831 lines from /Users/saki/Desktop/AnomalyDetection/graph/enron_edgelist.txt
Preprocessing progress: 99.75%
Walking Progress: 98.11%
Learning Progress: 99.99%
Sakis-MacBook-Pro:~ saki$
```

图 5-2：使用 node2vec 算法生成安然公司邮件数据集的图向量表征

然后通过 C++编程根据邻接表计算出所有节点的度作为补充特征，由于节点的度数值与 node2vec 算法求出的图向量表征数值相比较较大，所以这里对其进行归一化处理，将每一个节点的度除以所有节点的度的最大值，从而使这一维的数值控制在 0 到 1 的区间内，代码文件为 CalcDegree.cpp。另外由于 node2vec 算法的输出文件会将节点顺序打乱，所以需要通过对 SortByID.cpp 文件将所有节点按照 ID 的升序进行排序。

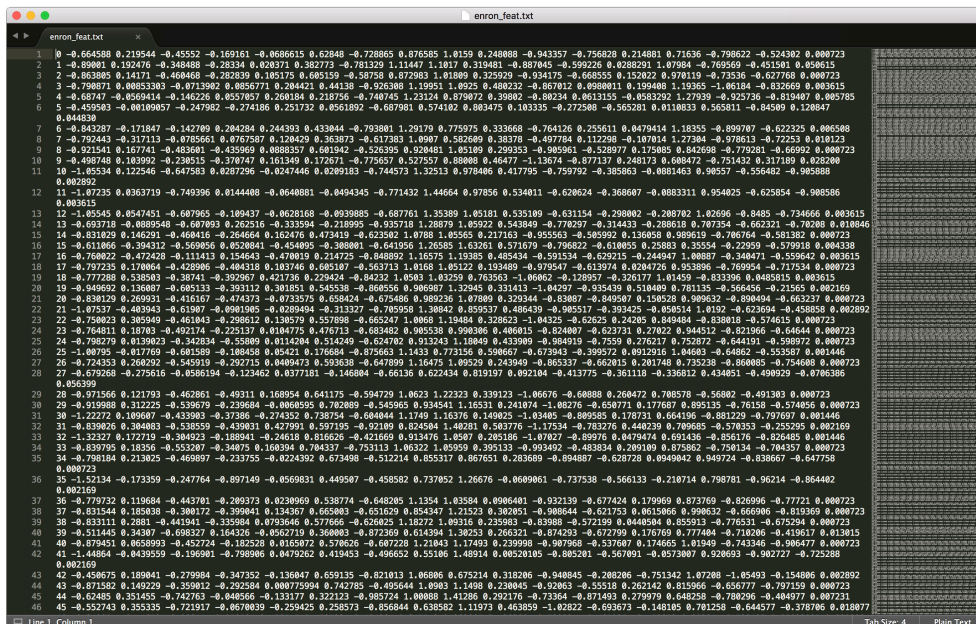


图 5-3：节点特征向量文件

最后生成所有节点的特征向量文件 `enron_feat.txt` 格式如图 5-3 所示，每一行第一个元素为节点 ID，后面为该节点的 17 维特征向量。

5.3.2 初始化节点聚类

为了减少迭代次数，我们首先使用 K 均值算法对所有节点进行聚类，生成初始社区。这里，我们使用的是 Python 机器学习库 `sklearn`⁷ 实现节点的聚类，代码文件为 `clustering.py`，生成的社区划分文件为 `enron_kmeans.csv`，其格式如图 5-4 所示，每一行为节点 ID 与社区编号，例如节点 0 属于第 11 个社区。

0	11
1	11
2	11
3	6
4	6
5	11
6	6
7	6
8	27

图 5-4：社区划分文件

⁷ <http://scikit-learn.org/stable/index.html>

5.3.3 学习隶属度向量

由于节点隶属度向量的迭代更新过程中涉及到大量的向量与矩阵运算，需要通过多重循环来实现，考虑到 Python 在时间效率上的缺点，所以我们选择使用 C++ 进行编程计算，代码文件为 CalcProb.cpp。代码中主要包括两个更新函数 updateM() 与 updateY()，以及 objFunc() 用于计算返回目标函数 O 的值，程序每次迭代会输出当前迭代次数以及目标函数的值，如图 5-5 所示。程序达到迭代次数后会将计算出的所有节点的隶属度向量输出至文件 enron_pr.txt 中。

```
Iteration time: 1
Updating M...
Updating Y...
Value of O: 311316.76626
Iteration time: 2
Updating M...
Updating Y...
Value of O: 207635.73453
Iteration time: 3
Updating M...
Updating Y...
Value of O: 199928.67561
```

图 5-5：学习隶属度向量过程中的输出

5.3.4 判断异常用户

在得到所有节点的隶属度向量后，我们通过 4.2.5 节中的公式 (4.19) 对所有节点的异常系数进行计算，之后再与人为设定的阈值 $thre$ 进行比较，超过阈值的节点即为异常用户，被选出后从邻接表文件和社区划分文件中删除，代码文件为 CalcAScore.py。

至此，数据集中的异常用户已被全部检测出来并删除。

5.4 实验结果与分析

本文中通过比较删除异常用户前后社交网络的模块度 (Modularity) ^[17] 来评估社区发现的质量的改进效果，从而评估检测算法的有效性。模块度也称模块化

度量值，是目前常用的一种衡量网络社区结构强度的方法，最早是由 Newman 提出^[11]，模块度 Q 的计算公式为：

$$Q = \sum_{c \in C} \left(\frac{I_c}{E} - \left(\frac{2I_c + O_c}{2E} \right)^2 \right) \quad (5.1)$$

其中， E 为图中的总边数， C 为图中的社区集合， I_c 为两个端点都在社区 c 中的边数， O_c 表示一个端点在社区 c 中，而另一个端点不在社区 c 中的边数。

模块度的大小主要取决于网络的社区划分情况，其值越接近于 1，表示网络划分的社区结构的强度越大，也就是社区划分的质量越好。一般来说，一个社交网络的模块度介于 0.3 与 0.7 之间。

计算模块度的代码文件为 CalcModularity.py，其输入包括一个图的邻接表文件以及一个社区分化的结果文件。

本次实验中人为设定的变量包括社区数量 k 、调节参数 α 、最大迭代次数 t_{max} 、异常系数阈值 $thre$ ，我们通过改变这些变量的值多次实验，并与删除异常用户之前的数据集的模块度对比，进行实验结果的分析。其中，由于目标函数 O 收敛速度较快，在迭代 10 次后就基本不再变化，所以这里我们不对 t_{max} 的值进行修改，所有的实验均是在 $t_{max} = 10$ 的情况下进行。同样，调节参数 α 我们的取值有{1,0.01,0.0001}，但是不同的取值对于模块度的影响并不明显，所以这里也不在折线图中显示。

当社区数量 k 分别取{70,50,40}时，异常系数阈值 $thre$ 与模块度 Q 的关系分别如图 5-6、图 5-7、图 5-8 所示。在上面每一张折线图中，横坐标为异常系数阈值 $thre$ ，纵坐标为模块度 Q ，蓝色折线表示本文的实验结果，红色水平直线作为

基准线表示在该社区数量下未删除异常用户的初始数据集的模块度。由于对于不同的社区数量，异常系数需要有不同的取值，所以这里分为三张图展示。

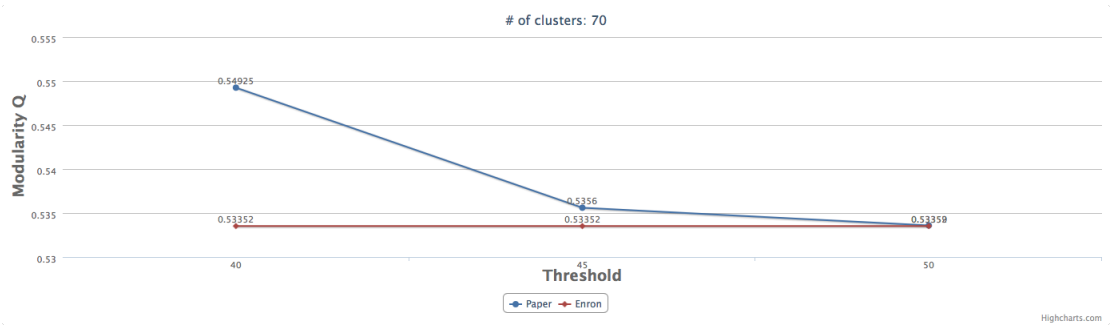


图 5-6：社区数量为 70

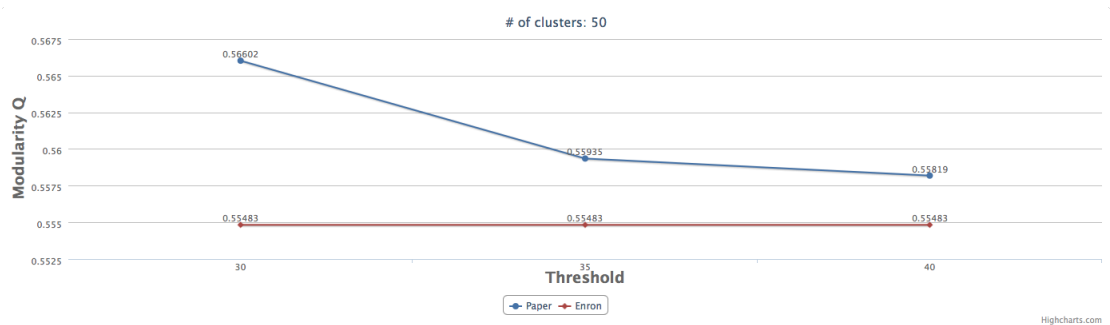


图 5-7：社区数量为 50

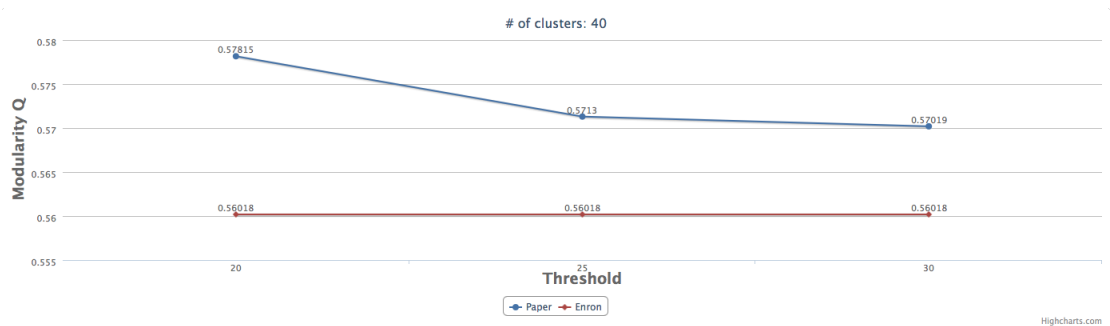


图 5-8：社区数量为 40

从以上结果中可以看出，在删除异常用户后，该网络的模块度均比之前有所提高。这是因为我们删除了异常用户后，使得在这个网络中，那些跨社区连接的边被删除，从而使得每个社区内部的连接更加紧密，所以模块度变大，这是符合模块度的定义的。

另外，可以看出在异常系数阈值 $thre$ 逐渐变大的过程中，该网络的模块度开始下降，这也是合理的，因为异常系数阈值 $thre$ 的取值决定了需要删除的用户数量，所以如果我们以从右向左的变化趋势来看这三张图，当异常系数阈值 $thre$ 的取值变大，也就意味着用户为正常用户的要求更加严格，从而被删除的异常系数较大的用户变多，这使得社区内部的连接更加紧密，而社区之间的连接则会更加稀疏，从而增大了模块度。

对于社区数量的三种取值，本文实验结果中的最大模块度值分别比初始数据集的模块度提高了 2.9%、2.0%、3.2%。所以通过本次实验，证明了本文设计的异常用户检测算法是确实有效的，我们从社交网络中找出了异常用户，提高了社区发现的质量。

第6章 结语

为了检测社交网络中存在的异常用户从而解决其对社交网络带来的问题，本文首先提出了一个对于异常用户的合理的定义，在此基础上提出了基于图向量表征的异常用户的检测算法。通过 Network Embedding 算法首先得到所有用户节点的特征向量，然后利用聚类分析的思想学习出每一个节点属于每一个社区的隶属度，最后利用节点的隶属度向量计算其异常系数从而判断该节点所代表的用户是否为异常用户。另外根据这个检测算法设计程序，使用公开的社交网络数据集进行实验，最后取得了较好实验结果，成功检测出了其中的异常用户，提高了网络的模块度。

当然，本文还存在着一定的不足，需要在日后的工作中加以改进：

(1) 本文的检测算法虽然能够成功找出存在于社交网络中的符合定义的异常用户，但是由于社交网络为无标度网络，在现实生活中会有少量用户拥有较大的度，例如在推特中存在像美国总统特朗普这样非常受欢迎受关注的用户，他们的确同时连接了多个社区的正常用户，然而在实际生活中我们并不会将他们归类于异常用户。所以在未来的工作中还需要进一步挖掘其他特征并改进算法，将这一小部分用户从异常用户中排除。

(2) 本文的检测算法比较依赖于开始时通过 Network Embedding 算法获得的用户节点特征向量，由于对 node2vec 算法的开源程序不够熟悉，所以在输入命令时对参数的配置还不够精确，从而可能使得 node2vec 算法的准确率有所降低。未来还需要在调参过程中慢慢尝试，找到最适用于相应的社交网络数据集和图向量表征目标的参数配置。

(3) 本文的检测算法中使用的社区这一概念，其实相对比较抽象主要用于将所有节点归入不同的类别中，所以人为设定的社区数量并不一定等于该社交网络中实际的社区数量。至于该社交网络中实际究竟有多少个社区，这里留作未来的工作去研究。

参考文献:

- [1]. 江东灿, 陈维政, 闫宏飞. 基于 DeepWalk 方法的适应有限文本信息的 DWLTI 算法. 2017.
- [2]. 莫倩, 杨珂. 网络水军识别研究. 软件学报, 2014: 1505-1526.
- [3]. 张玉清, 吕少卿, 范丹. 在线社交网络中异常账号检测方法研究. 计算机学报, 2015.
- [4]. Akoglu L, McGlohon M, Faloutsos C. Anomaly Detection in Large Graphs. 2009.
- [5]. Akoglu L, Tong H, Koutra D. Graph-based anomaly detection and description: A survey. Data Mining and Knowledge Discovery, 2014: 1-17.
- [6]. Bhat S Y, Abulaish M. Community-Based Features for Identifying Spammers in Online Social Networks. ASONAM. 2013.
- [7]. Caviglione L, Coccoli M, Merlo A. A taxonomy-based model of security and privacy in online social networks. International Journal of Computational Science and Engineering, 2014: 325-338.
- [8]. Chandola V, Banerjee A, Kumar V. Anomaly Detection: A Survey. 2007
- [9]. Chu Z, Gianvecchio S, Wang H, et al. Who is tweeting on Twitter: Human, bot, or cyborg. 2010.
- [10]. Cui P. Network Representation Learning: A Revisit in the Big Data Era. KDD. 2016.
- [11]. Clauset A, Newman M, Moore C. Finding community structure in very large networks. Physical Review E, 70(6): 66111. 2004.
- [12]. Cristofaro E D, Friedman A, Jourjon G, et al. Paying for likes: Understanding facebook like fraud using honeypots. 2014.
- [13]. Ellison N B. Social network sites: Definition, history, and scholarship. Journal of Computer-Mediated Communication, 2007: 210-230.
- [14]. Fire M, Goldschmidt R, Elovici Y. Online social networks: Threats and solutions survey. IEEE

Communications Surveys and Tutorials, 2013: 2019-2036.

- [15]. Gao H, Hu J, Huang T, et al. Security issues in online social networks. IEEE Internet Computing, 2011: 56-63.
- [16]. Grover A, Leskovec J. node2vec: Scalable Feature Learning for Networks. KDD, San Francisco, CA, USA. 2016.
- [17]. Hu R, Aggarwal C C, Ma S, et al. An Embedding Approach to Anomaly Detection. ICDE. 2016.
- [18]. Kanich C, Kreibich C, Levchenko K, et al. Spamalytics: An Empirical analysis of spam marketing conversion. 2008.
- [19]. Liu Y, Chawla S. Social Media Anomaly Detection: Challenges and Solutions. WSDM. 2017.
- [20]. Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space. CoRR. 2013.
- [21]. Perozzi B, Al-Rfou R, Skiena S. DeepWalk: Online Learning of Social Representations. 2014.
- [22]. Savage D, Zhang X, Yu X, et al. Anomaly detection in online social networks. Social Networks. 2014: 62-70.
- [23]. Shrivastava N, Majumder A, Rastogi R. Mining (Social) Networks Graphs to Detect Random Link Attacks. ICDE. 2008.
- [24]. Stringhini G, Wang G, Egele M, et al. Follow the green: Growth and dynamics in Twitter follower markets. 2013.
- [25]. Tang J, Qu M, Wang M, et al. LINE: Large-scale Information Network Embedding. WWW, Florence, Italy. 2015.
- [26]. Thomas K, McCoy D, Grier C, et al. Trafficking fraudulent accounts: The role of the underground market in Twitter Spam and abuse. 2013.

- [27]. Wang D, Cui P, Zhu W. Structural Deep Network Embedding. KDD, San Francisco, CA, USA. 2016.
- [28]. Xue J, Yang Z, Yang X, et al. VoteTrust: Leveraging friend invitation graph to defend against social network sybils. 2013.

致谢

本文的工作是在中山大学 Inpluslab 的郑子彬和吴嘉婧两位老师以及叶方华师兄的指导下完成的。感谢他们从选题到成文期间多次给我提供耐心细致的指导讲解以及宝贵建议，让我可以顺利完成本文。在此向两位老师以及叶方华师兄表示衷心的感谢。

毕业论文是本科期间的最后一次作业，更是对四年学习的汇报总结。感谢大学四年我的所有授课老师，是你们的辛苦付出让我打下了本文的知识基础，也让我坚定了在学习计算机科学这条道路上继续学习下去的决心。其中，特别感谢实验室的郑子彬老师，作为我之前数据挖掘课程的授课老师，让我对这个领域产生了浓厚的兴趣。老师还给予了我后来在实验室实习的机会，让我掌握了许多数据挖掘方面的本领，也接触了很多前沿流行的技术。在此向所有老师致以崇高的敬意以及衷心的感谢。

感谢大学四年我的所有同学、朋友，是他们在学业、生活上对我的帮助，让我能够顺利的完成本科的所有课程，让我能够在广州这座陌生的城市留下一段美好难忘的生活经历。

最后要感谢的是我的父母、家人们，作为我成长过程中最坚实的后盾，一直默默的为我提供源源不断的支持和鼓励，感谢他们这么多年以来对我无微不至的关爱和照顾。

毕业论文（设计）成绩评定记录 Grading Sheet of the Graduation Thesis (Design)

指导教师评语

Comments of Supervisor:

按计划完成，完成情况优。

成绩评定

Grade:

指导教师签名

Supervisor Signature :

Date:

答辩小组或专业负责人意见

Comments of the Defense Committee:

成绩评定

Grade:

签名:

Signatures of Committee Members

Date:

院系负责人意见

Comments of the Academic Chief of School:

成绩评定

Grade:

签名

Signature:

院系盖章

Stamp:

Date: