虚假用户检测

《社交网络分析》课程实践项目报告

2021-1-15

队长：郭家兴　学号：20210980101
队员：曹恺燕　学号：20210980133
队员：赵月涓　学号：20210980144

1 问题背景

在过去的几年里，在线社交网络迅猛发展，如 Facebook、Twitter、微博等，已经成为互联网用户之间沟通与交流的主要方法之一。然而，随着技术和商业上的巨大成功，社交网络平台也为广播垃圾邮件的发送者提供了大量的机会去传播恶意消息和行为。垃圾邮件主要指的是包含未经请求的包含恶意链接的消息，一般由虚假账号发出，它将其他真实用户引导到包含恶意软件下载、钓鱼、药品销售或骗局等的外部网站，造成不利影响。微博作为中国最大的社交网站之一，每天都吸引着数以百万的在线用户，为了增加用户良好体验，进行虚假用户检测显得尤为重要。针对这个问题，我们分别使用了有监督学习方法和无监督学习方法进行虚假用户检测。

2 虚假用户识别有监督学习

2.1 微博数据集介绍

我们使用 Python 爬取了 900 位微博网友的个人信息以及他们之间的连边信息，经过筛选，其中 791 位为正常用户，109 位为虚假用户。爬取个人数据包含 id、昵称、性别、地区、微博数、粉丝数、关注数、个人简介等共 11 个字段。

2.1.1 网络特征性质

我们首先对微博用户网络进行描述性分析，初步刻画其关系网络特征。

从统计结果看出，在微博网络中，由于关注关系具有较高的传递性，网络的平均聚类系数是高于 web 平均聚类系数 0.081 的。其次，微博网络中两个节点的平均路径长度较短，呈现“小世界现象“。并且在微博中，人们除与自己的现实生活中好友进行关注外，还会关注一些知名度较高的用户，这样网络里存在很多边的两个节点度差别很大，这样网络的通配系数是负的。此外，网络的密度较小，网络比较稀疏。

微博网络特征描述	
平均度	2.973
聚类系数	0.191
同配系数	-0.342
网络密度	0.003
度中心性	0.006
网络直径	10

表 1：微博网络特征描述

2.1.2 微博网络图

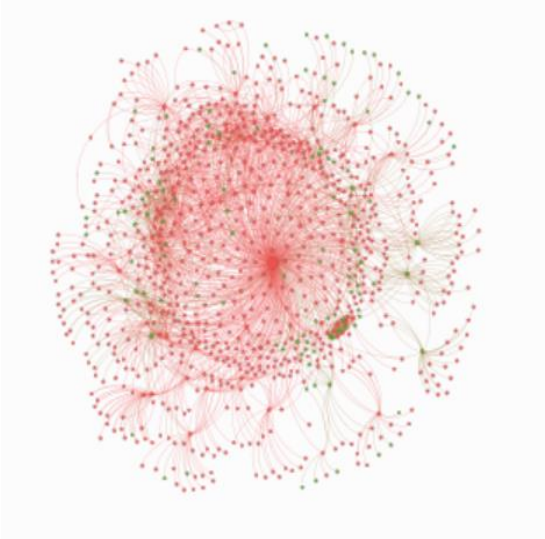


图 1：网络图

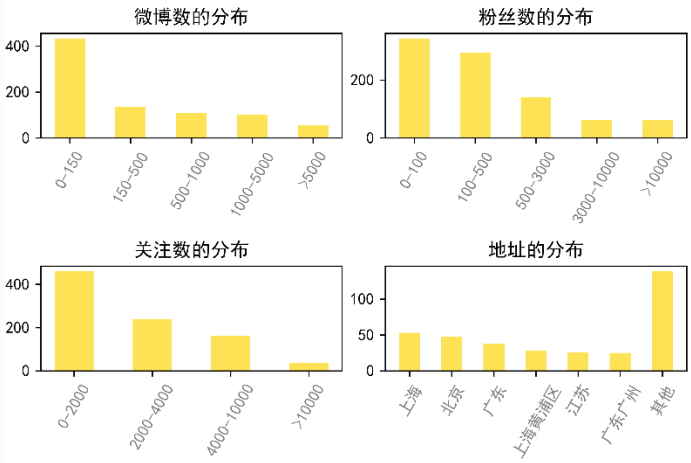


图 2：网络特征

图中不同的节点代表不同的用户，节点的大小代表用户的度，节点的颜色代表用户的属性，红色的是正常的用户，正常用户占大多数，绿色的是虚假用户，在网络图中可以看到超级节点的存在，即部分用户获得了大多数用户的关注，图 2 中可以看出微博数、粉丝数、关注数分布也基本符合幂律分布，也说明微博网络符合真实世界无标度网络的特征。

2.2 虚假用户和真实用户网络异同分析

我们提取出只有真实用户和只有虚假用户的网络，在 Gephi 中根据 Fruchterman Reingold 算法绘制网络图像，并且计算两种网络的特征进行分析比较。

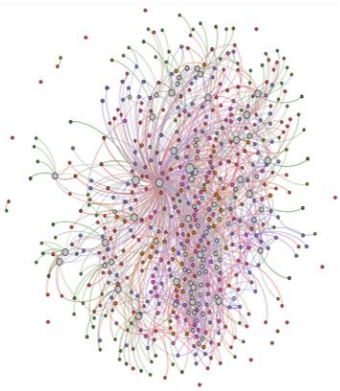


图 3：正常用户网络

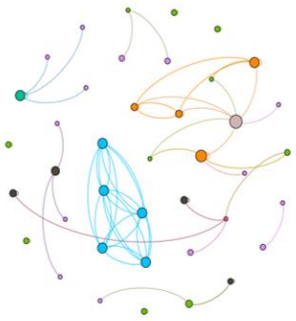


图 4：虚假用户网络

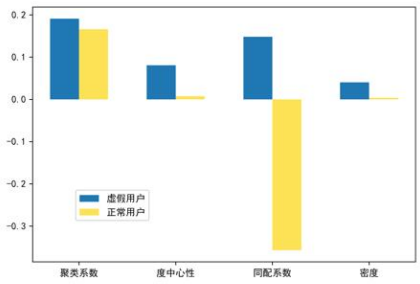


图 5：特征比较

网络图中的颜色代表入度的大小，灰色、蓝色、橘色、红色、绿色的入度依次减少，节点的大小代表节点的度。可以看出正常用户网络中三元闭包数目是少于虚假用户网络的，虚假用户相互关注，形成抱团的趋势。

从图中可以看出，真实用户和虚假用户的聚类系数是相似的。而在真实网络里，人的社交能力是有上限的，所以不会关注特别多的人，所以真实网络里度的

中心性不会特别高。同配系数用来评价度相近的节点是否倾向于互相连接，在网络中真实用户一般会关注好友或者比较有名的用户，这样就会有很多边两端节点度差异很大，同配系数是负的。真实用户的网络比较稀疏，而虚假用户的网络比较稠密，这与虚假用户相互抱团也有关系

2.3 有监督学习方法

2.3.1 解决模型框架

其实该虚假用户检测问题可以看作是一种二分类问题，因此传统机器学习分类方法都可应用其中，在此，我们采取了支持向量机、决策树、随机森林和极端梯度提升树以及 LSTM 分类方法解决这个问题。

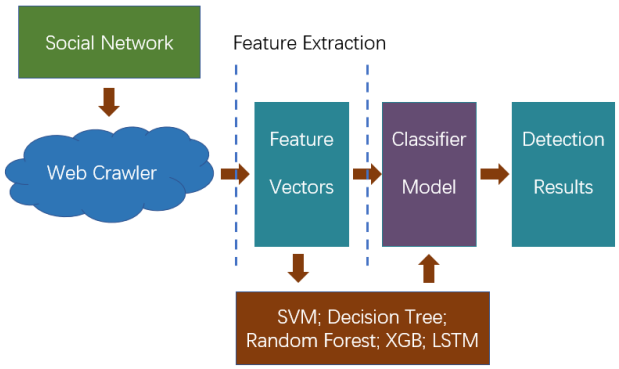


图 1： 垃圾邮件检测模型概述

2.3.2 传统机器学习方法

2.3.2.1 支持向量机 SVM

支持向量机（support vector machines, SVM）是一种二分类模型，它的基本模型是定义在特征空间上的间隔最大的线性分类器；其还包括核技巧，这使它成为实质上的非线性分类器。SVM 学习的基本想法是求解能够正确划分训练数据集并且几何间隔最大的分离超平面，可形式化为一个求解凸二次规划的问题，也等价于正则化的合页损失函数的最小化问题。

$$\max_{\alpha} \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j k(x_i, x_j)$$

$$s.t. \sum_{i=1}^{m} \alpha_i y_i = 0,$$

$$\alpha_i \geq 0, \quad i = 1,2, \ldots m.$$

求解后可得到 $f(x) = \sum_{i=1}^{m} \alpha_i y_i k(x_i, x_j) + b.$

$k(\cdot,\cdot)$为选择的核函数，我们采用的是非线性 SVM 分类器径向基函数核(RBF)。

2.3.2.2 决策树分类

决策树（ Decision Tree ）又称为判定树，是运用于分类的一种树结构。其使用层层推理来实现最终的分类。一般地决策树右一个根节点、多个内部节点和叶节点组成。

2.3.2.3 随机森林分类器

随机森林属于集成学习中的 Bagging 方法，其是由很多决策树构成的，不同决策树之间没有关联。当我们进行分类任务时，森林中的每一棵决策树分别进行判断得到分类结果。而后随机森林就会把分类最多的决策树的分类结果作为最终的结果。

其学习算法包括四个步骤：随机抽样训练决策树、随机选取属性做节点分裂属性、重复步骤 2 直到不能再分裂、建立大量决策树形成森林。

2.3.2.4 XGB 分类器

XGB 中文名称为极端梯度提升树，使用 CART 回归树或线性分类器作为基学习器，是一种 boosting 算法。其由多个相关联的决策树联合决策，即下一棵决策树输入样本会与前面决策树的训练和预测相关。

2.3.3 深度学习方法

2.3.3.1 长短期记忆神经网络（LSTM）

长短期记忆网络（LSTM）是循环神经网络的一个变体，可以有效地解决简单循环神经网络的梯度爆炸或消失问题。

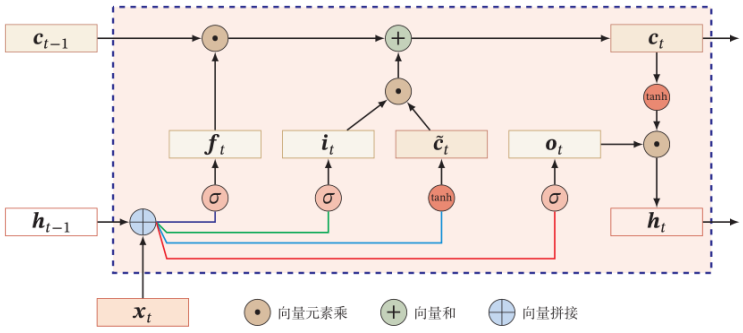


图 3： LSTM 状态图

在 $h_t = h_{t-1} + g(x_t, h_{t-1}; \theta)$的基础上，LSTM 网络主要改进在以下两个方面：新的内部状态 LSTM 网络引入一个新的内部状态 $c_t \in R^D$ 专门进行线性的循环信息传递，同时（非线性地）输出信息给隐藏层的外部状态$h_t \in R^D$. 内部状态 c_t 通过下面公式计算：

$$c_t = f_t \odot c_{t-1} + i_t \odot \widetilde{c_t},$$

$$h_t = o_t \odot \tanh(c_t),$$

其中 $f_t \in [0,1]^D$、$i_t \in [0,1]^D$、$o_t \in [0,1]^D$分别为遗忘门、输入门和输出门来

控制信息传递的路径；⊙为向量元素乘积；c_{t-1}为上一时刻的记忆单元；$\widetilde{c_t} \in R^D$是通过非线性函数得到的候选状态：

$$\widetilde{c_t}, = \tanh(W_c x_t + U_c h_{t-1} + b_c).$$

在每个时刻t，LSTM 网络的内部状态 c_t记录了到当前时刻为止的历史信息。

2.3.3.2 深度学习方法

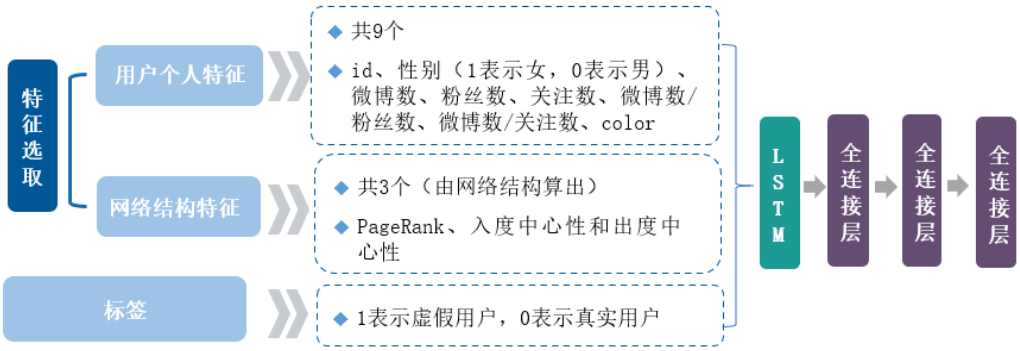


图 3 深度学习方法

本文将所有的特征输入 LSTM 层后通过三层全连接层，最后得到分类结果。

2.4 实验

2.4.1 特征结果选取

如图所示，我们选取了共 12 个特征包括 id、性别、微博数、粉丝数、关注数等 9 个用户个人特征和 PageRank、入度中心性和出度中心性等 3 个网络结构特征，将其输入到模型当中学习，而后得到分类结果。

2.4.2 分类结果比较

在 python 当中像 SVM、决策树、随机森林以及 XGboost 分类器都可以直接调用 sklearn 包，为了平衡样本，均采用了 class_weight=”Balanced”，其它设置均为默认参数; 深度学习方法中，我们输入用户的 12 个个人特征，首先通过一层 LSTM 层，而后通过三层全连接层，即可得到输出结果。本文使用准确率、召回率和 F-score 来作为模型的评价指标，如表 1 所示，一方面在传统机器学习方法中，表现最好的是随机森林分类器，真实用户预测分类结果的精确率、召回率和 F-score 分别高达 100.00%、96.30%和 98.11%，虚假用户预测分类结果的精确率、召回率和 F-score 分别为 76.92%、100.00%和 86.96%；另一方面，LSTM 方法表现出来的效果不如传统的机器学习方法，这也许与它更适用于时序数据的训练有关。从测试集来看，如表 2 所示随机森林预测分类结果高达 96.70%。

表 1 虚假用户识别有监督学习结果（精确率、召回率和 F-Score）

Classifier	Precision		Recall		F-measure	
	spammer	Non-spammer	spammer	Non-spammer	spammer	Non-spammer
SVM	50.00%	98.63%	90.00%	88.89%	64.29%	93.51%
Decision Tree	71.43%	100.00%	100.00%	95.06%	83.33%	97.47%
Random Forest	76.92%	100.00%	100.00%	96.30%	86.96%	98.11%
XGB Classifier	71.43%	100.00%	100.00%	95.06%	83.33%	97.47%
LSTM	64.29%	98.70%	90.00%	93.83%	75.00%	96.20%

表 2 虚假用户识别有监督学习结果(测试集准确率)

Classifier	SVM	Decision Tree	Random Forest	XGB	LSTM
Accuracy for test	89.01%	95.60%	96.70%	95.60%	93.41%

3 无监督学习

本文基于 SybilBlind 算法进行虚假用户的无监督学习[1]。

3.1 无监督学习的必要性

（1）针对一个大的数据集，手动标注虚假用户非常耗费时间。

（2）有监督学习需要基于大量的特征识别虚假用户，而这些特征是随时间变化的且可人工反检测，因而有监督学习模型不具有良好的时效性。

3.2 本文无监督学习基本假设

网络属于同质网络（homophily network），即该网络中任一条连边的两个端点大概率是同类用户（同为正常用户或同为虚假用户）。

3.3 SybilBlind

首先从整个网络中选取一小部分用户节点（如整个网络总共有 1000 个节点，可以选出 20 个节点），随机分为相等的两组 B 和 S，对其中 B 全部标为正常用户，另一组 S 全部标为虚假用户。设n_{bb}表示 B 中的正常用户的个数，n_{bs}表示 B 中的虚假用户个数，n_{sb}表示 S 中的正常用户个数，n_{ss}表示 S 中的虚假用户个数。然后我们定义三种极化如下

正极化：$n_{bb} > n_{sb}$且$n_{bs} < n_{ss}$

负极化：$n_{bb} < n_{sb}$且$n_{bs} > n_{ss}$

非极化：$n_{bb} = n_{sb}$且$n_{bs} = n_{ss}$

不难发现，在正极化情况下，我们给定的大多数随机初始标签与用户真实标签相同；在负极化情况下，我们给定的大多数随机初始标签与用户真实标签相反；在非极化情况下，给定的随机初始标签一半正确，一半错误。

注意，在给定随机初始标签之后，我们使用基于网络结构和随机游走的SybilSCAR[2]算法对其他用户节点做出分类判断，在同质网络的假设下，若发生负极化，则意味着经过足够多的随机游走迭代次数之后，原来网络中的绝大多数正常用户节点会被判断为虚假用户节点，这是我们所不希望发生的；反之，我们希望的是正极化的情况。下面，我们根据预测标签网络的同质性和熵两个指标对以上所提及的三种极化情况做出识别，以便我们选择正极化的结果。

定义同质性h和熵 e 如下

$$\mathrm{h} = \frac{\#homogeneous}{\#edges\ in\ total}$$

$$\mathrm{e} = \begin{cases} 0, & \mathrm{s} > 0.5 \\ -\mathrm{slog(s)} - (1-\mathrm{s})\log(1-\mathrm{s}), & \text{otherwise} \end{cases}$$

其中，$\#homogeneous$表示网络中两个端点的预测类别相同的边的个数，$\#edges\ in\ total$为网络中总的边数，s表示网络中被预测为虚假用户的用户所占的比例。由于现实中的社交网络往往是正常用户多于虚假用户（如 Twitter 网络中虚假用户大约只占 10%），故当负极化发生时，由于多数正常用户会被预测为虚假用户，故最终预测为虚假用户的节点会超过 50%，也即$\mathrm{s} > 0.5$。这样我们就可以通过$\mathrm{e} = 0$识别出这种负极化的情况；而当非极化发生时，节点标签的预测更接近于随机预测，因而同质性会很差，所以我们可通过同质性 h 特别小识别出非极化的情况；而当同质性 h 和熵 e 都比较大时，就是我们所希望的正极化情况。

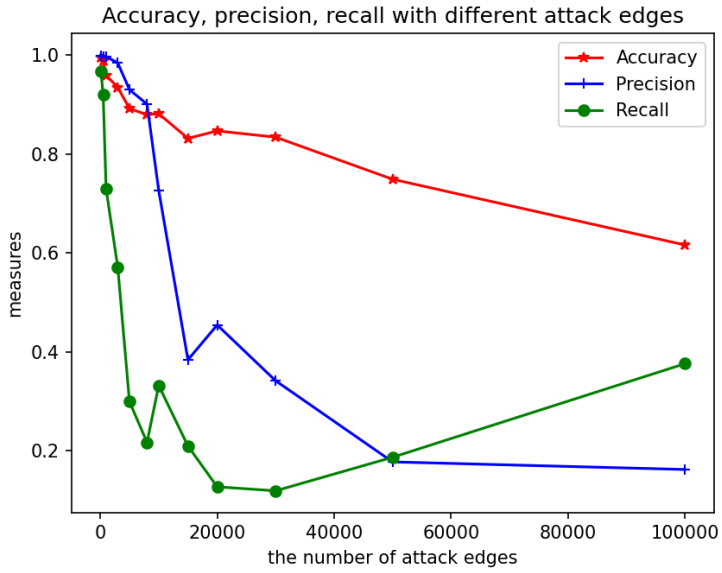
事实上，我们会重复做大量随机初始标签的实验，然后首先从中选出同质性前 k 大的实验，再从这 k 个样本中选择 e 最大的那一次实验的预测结果作为我们无监督学习的预测结果。这样得到的预测结果通常对应的是正极化中能够检测出最多虚假用户的结果（熵$\mathrm{e} \geq 0$，且随 s 在$(0, \frac{1}{2})$上单调递增，在$(\frac{1}{2}, 1)$上单调递减）。

我们再来分析一下 SybilBlind 算法的时间复杂度。在每次实验的每轮迭代中，需要从一个节点出发遍历其所有邻居节点，然后再遍历其邻居节点的邻居节点……平均时间复杂度为$\mathrm{O}(|\mathrm{E}|)$。因而若一共进行N次重复实验，每次实验随机游走迭代T步，总共时间复杂度就是$\mathrm{O}(\mathrm{KT}|\mathrm{E}|)$。

3.4 实验验证

我们选取 soc-Epinions 数据集的前五万个用户节点极其之间的连边（约 46 万条连边），全部标为正常用户；然后，通过优先链接（PA）模型生成一个具

有 8790 个节点及其连边的网络（约 35000 条连边），全部标为虚假用户；最后添加“attack edges”，attack edges 即表示正常用户和虚假用户之间的连边（由于同质网络的假设，要注意不能生成过多 attack edges）。我们分别添加 100、500、1000、3000、5000、8000、10000、15000、20000、30000、50000、100000 条 attack edges，将预测结果得到的 accuracy、precision 以及 recall 指标绘制成三条随 attack edges 数量变化的曲线如下：



不难发现，当 attack edges 个数小于 10000 时，该网络保持有比较好的预测性能，可认为该网络属于同质网络；而当 attack edges 个数大于 10000 时，该网络对虚假用户的预测性能变得很差，可认为此时其不再满足同质性假设，从而不再适用于 SybilBlind 算法。由此，我们也可以了解到当网络中异质边占所有边的比例小于 2%时，可认为该网络为同质网络；否则应认为该网络不是同质网络。

4 总结

在数据集有标签的情况下，构建关于网络结构和节点信息两方面的特征，随机森林分类器效果最好、最可信。

在数据集无标签的情况下，采用 SybilBlind 算法，重复多次实验，并用同质性和熵两个指标将结果进行聚合，在网络满足同质性假设的情况下可得到高可信度的分类结果。

5 参考文献

[1] Wang B , Zhang L , Gong N Z . SybilBlind: Detecting Fake Users in Online Social Networks without Manual Labels[C]// International Symposium on Research in Attacks, Intrusions, and Defenses. Springer, Cham, 2018.
[2] Wang B , Jia J , Zhang L , et al. Structure-based Sybil Detection in Social Networks via Local Rule-based Propagation[J]. IEEE Transactions on Network Science & Engineering, 2018, PP(99):1-1.