电商产品评论数据的异常检测

孙澈：2120171054 黄文举：2120171019
程显达：2120171005 姜玮：2120171021

摘 要

随着电子商务的发展，评论信息对于消费者的导向作用越来越大，其中虚假的异常评论会对消费者或商家造成不良的影响，因此从海量数据中检测出虚假的异常评论是一个值得研究的话题。电商产品的评论数据是海量的，其中异常数据只占有一小部分，极端不平衡的样本分布使得采用分类方法检测异常变得困难。为了应对这个问题，本文提出了一种基于自编码的深度高斯混合模型的离群点检测方法，在训练过程中只学习正常的评论特征：先使用深度自编码网络对其进行降维，然后采用深度高斯混合模型对正常样本进行聚类。在测试时，距离所有聚类中心都比较远的点认为是离群点，即不同于正常样本的异常评论数据，完成异常检测任务。本文的方法在 YelpCHI 数据库上取得了不错的效果。

1 背景

随着互联网的发展和普及，基于互联网开展的电子商务已经成为人们进行商务活动的新模式，随之而来的是网络上充斥着的大量的关于产品的评论信息，这些信息对消费者选择购物有重要的导向作用，一些异常的评论信息（包括虚假的好评信息以及恶意的差评信息）已经严重影响到消费者的用户体验和切身利益，因此，如何检测出异常的评论是一个值得研究的热点问题。

现在已经有不少方法可以进行异常评论过滤，其中最常用的方法是提取评论的行为特征和内容特征进行分类，来区分正常和异常评论，取得了不错的效果，例如贝叶斯分类器和支持向量机（svm）分类器等。除了分类的方法外，还有根据异常评论发布者的行为特征以及评论内容的情感特征等进行数据挖掘（如过于偏激的打分、频繁的发布评论和极端的描述等特征），挖掘这些特征发现异常评论的出现规律，然后完成检测任务。但是，虽然异常的评论数量也不算少，但是和大量的可获得的正常的评论相比，数量还是相对过少，样本分布极其不均衡的数据使得分类存在一定的问题，同时数量不足的异常样本也使得发现异常样本的内在规律变得困难，并且异常样本也不是一成不变的，因此上面提到的两个方法

都具有一定的局限性。

本文考虑到上面提到的问题，将异常评论看成离群点，采用离群点检测的方法训练检测模型并检测异常评论。本文在训练检测模型的时候只使用到了正常的评论，学习正常评论的内在规律和模式，然后在测试时通过离群点检测来检测异常。这种建模正常评论、然后通过离群点检测的方法检测异常的方法避免了因样本分布不均匀而产生的问题，也考虑到了异常样本数量相对较少的特点，可以很好地完成检测任务。我们的方法是先使用深度自编码网络[1]对正常评论的特征进行降维，然后用深度高斯混合[2]模型对降维后的特征进行聚类，在测试阶段，正常的评论可以很好地聚在我们在训练时得到的聚类中心处，而异常评论则距离所有的聚类中心都比较远，达到检测异常评论的目的。下面是我们方法的贡献：

（1）提出了基于自编码的深度高斯混合模型，仅在正常评论上做训练，可以建模正常样本的特征并进行聚类，在测试时将异常评论视为离群点，如果样本到所有聚类中心的距离相对都比较大，则认为这是离群点（异常评论），完成异常评论检测的任务。

（2）在训练模型时，仅仅使用正常评论的样本，这些样本是很容易得到的且是很多的，满足深度网络对大数据的需求，这可以很好地建模正常评论，从而提高检测结果。

（3）利用深度自编码网络和深度混合高斯模型完成对正常样本的降维和聚类，深度网络在机器学习上的优越性能保证了本文方法的有效性。

本文接下来的部分构成： 第 2 章环境部分介绍了本文实验所需要的配置环境，第 3 章方法部分介绍了本文的基于自编码的深度高斯混合模型，第 4 章数据集介绍和预处理部分介绍我们使用的 YelpCHI 数据库和在该数据库上做的处理，第 5 章实验设置部分具体介绍实验中使用到的模型参数，第 6 章实验结果部分介绍了本文方法在检测异常评论上的效果，第 7 章结论部分总结本文工作。

2 环境

（1）操作系统：Widows 7

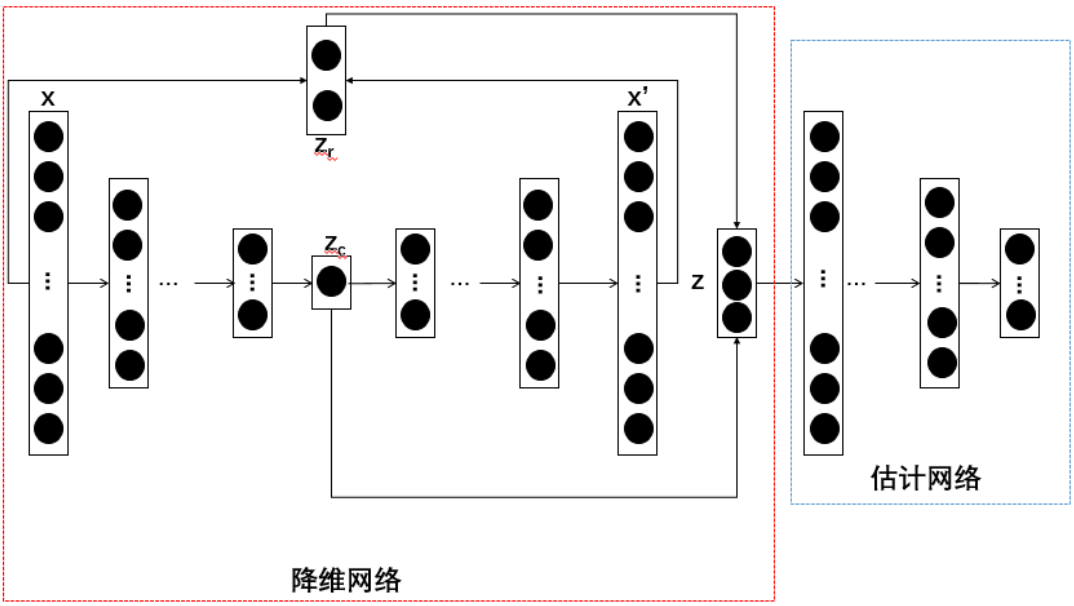
（2）语言和环境：Matlab R2016a 和 python 2.7. 其中 Python 所需要的工具

包：tensorflow, numpy, os, shutil, time, linecache, math, codecs, string, csv, sys, re, nltk, nltk.tokenize.punkt

（3）深度学习框架 tensorflow 1.2

3 方法

基于自编码模型的深度高斯混合模型包括两个主要的部分：一个由深度自编码网络组成的降维网络和一个由深度高斯混合模型组成的估计网络（用于聚类），如下图所示，左侧红色部分是用于降维的深度自编码网络，输入评论的特征表示，产生的中间隐变量和重构误差拼接起来作为下一个聚类网络的输入；右侧蓝色部分是深度高斯混合模型网络，接受上一步的输出作为输入，完成高斯混合模型的聚类任务。



3.1 降维网络（深度自编码模型）

自编码模型有两部分组成：编码部分和解码部分，编码部分将原始输入进行压缩降维，得到中间的隐变量，解码部分由中间的隐变量生成原始输入的重构输出，它是一种有效地无监督降维方法，目的是保留更为重要的信息，使得重构的输出和原始输入接近。

自编码模型的输出是评论的低维的表达，该表达由两个部分组成：（1）深度自编码网络的中间隐变量z_c；（2）自编码网络的重构误差z_r，假定输入x，那么低维表示为：

$$z_c = h(x;\theta_e),$$

$$x' = g(z_c;\theta_d),$$

$$z_r = f(x, x'),$$

$$z = [z_c, z_r],$$

这里z_c是降维后的表示，z_r是自编码的重构误差，θ_e和θ_d是深度自编码网络的参数，x′ 是x的重构，h(·)表示编码函数，g(·)是解码函数，f(·)是计算重构误差函数，在本文中，我们采用余弦距离和欧氏距离两种度量误差的函数，z_r是两个维度的。z是聚类网络（深度高斯混合模型）的输入。

3.2 估计网络（深度高斯混合模型）

接收降维网络的输出作为输入，然后使用深度高斯混合模型进行聚类分析。

首先简单介绍高斯混合模型（GMM）。GMM 模型是认为数据在样本空间中是由多个高斯分布混合而成，它目的是找出这些高斯分布的均值和方差，并将每一个样本对应到相应的高斯分布的均值点处，完成聚类任务。

因此每个样本可以用下面的公式表示

$$p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(|\mu_k, \Sigma_k),$$

这里$\mathcal{N}(x|\mu_k, \Sigma_k)$表示混合模型中第 k 个分量，$\pi_k$是混合系数且所有的$\pi_k$加和为1，可以认为是每个分量所占的比重。

GMM 常用于聚类。如果要从 GMM 的分布中随机地取一个点的话，实际上可以分为两步：首先随机地在这 K 个分量（聚类中心）之中选一个，每个分量被选中的概率实际上就是它的系数π_k，选中分量之后，再单独地考虑从这个分量的分布中选取一个点就可以了——这里已经回到了普通的高斯分布，转化为已知的问题。将 GMM 用于聚类时，假设数据服从混合高斯分布（Mixture Gaussian Distribution），那么只要根据数据推出 GMM 的概率分布来就可以了；然后 GMM 的 K 个分量实际上对应 K 个聚类中心。根据数据来推算概率密度通常被称作密度估计。特别地，当已知（或假定）概率密度函数的形式，而要估计其中的参数的过程被称作参数估计，在 GMM 中就是这样，概率密度函数的形式已知，需要估计出 GMM 中的参数(π_k, μ_k, Σ_k)，常用的方法是 EM 算法，这里不做介绍。

深度高斯混合模型在训练阶段，产生多个聚类中心ϕ，对于每一个样本采用多层感知器的方法预测其属于哪一个聚类中心。假定输入为z，预先定义的聚类中心的个数是 K，估计网络通过下面的公式做预测

$$p = MLN(z, \theta_m),$$

$$\hat{y} = soft\max(p),$$

这里ŷ是用于预测的 K 维度的向量，p是多层感知器的输出，θ_m是参数。在训练

时，我们一次输入 N 的样本用于聚类，它们所属的聚类中心的预测 k，那么可以通过下式估计 GMM 的参数：

$$\hat{\phi}_k = \sum_{i=1}^{N} \frac{\hat{\gamma}_{ik}}{N},$$

$$\hat{\mu}_k = \frac{\sum_{i=1}^{N} \hat{\gamma}_{ik} z_i}{\sum_{i=1}^{N} \hat{\gamma}_{ik}},$$

$$\hat{\Sigma}_k = \frac{\sum_{i=1}^{N} \hat{\gamma}_{ik} (z_i - \hat{\mu}_k)(z_i - \hat{\mu}_k)^T}{\sum_{i=1}^{N} \hat{\gamma}_{ik}},$$

这里$\hat{\gamma}_i$是对每个输入样本属于哪个每个高斯分布（聚类中心）的预测，$\hat{\phi}_k, \hat{\mu}_k, \hat{\Sigma}_k$分别是第 k 个高斯分布的所占比重（可能出现的概率值），均值和方差。

根据得到的参数，可以进行似然估计，最好的参数会使得似然估计最大，为了便于优化计算，我们取对数似然函数的负值作为能量损失函数，来指导网络的参数学习。

$$E(z) = -\log\left(\sum_{k=1}^{K} \hat{\phi}_k \frac{\exp\left(-\frac{1}{2}(z-\hat{\mu}_k)^T \hat{\Sigma}_k^{-1}(z-\hat{\mu}_k)\right)}{\sqrt{|2\pi\hat{\Sigma}_k|}}\right),$$

这里$|\cdot|$表示矩阵的行列式

在测试阶段，可以直接用使用上式计算能量函数，将具有较高输出能量的样本视为异常样本（似然函数值低，不属于该 GMM 中任何一个聚类中心的分布），通过选取阈值的方法可以有效地检测出异常评论。

3.3 目标函数

在给定的 N 个训练样本后，总的目标函数是

$$J(\theta_e, \theta_d, \theta_m) = \frac{1}{N}\sum_{i=1}^{N} L(x_i, x_i') + \frac{\lambda_1}{N}\sum_{i=1}^{N} E(x_i) + \lambda_2 P(\hat{\Sigma}),$$

该目标函数有三个部分组成:

$L(x_i, x_i')$是深度自编码网络的重构损失函数，如果重构损失足够小，重构出来的数据和原始数据就很接近，那么就认为中间的隐变量就保留了足够的有用信息，这里采用平方误差函数（向量的 L2 范数），即$L(x_i, x_i') = ||x_i - x_i'||_2^2$

$E(z)$建模输入样本属于当前 GMM 模型的概率，当能量函数足够小的，认为 GMM 的似然函数值足够大，表示当前的样本能够很好地被 GMM 模型表示。

为了避免模型的奇异性问题（当协方差矩阵的对角项退化为 0 时会导致只有平凡解），我们添加了惩罚项$P(\hat{\Sigma}) = \sum_{k=1}^{K}\sum_{j=1}^{d} \frac{1}{\hat{\Sigma}_{kij}}$，这里 d 表示低维特征 z 的维

度。

公式中的λ_1,λ_2是超参数，在这里我们选择λ_1=0.1，λ_2=0.005。

4 数据集介绍和预处理

4.1 Yelp 数据集

为了得到数据标签（正常还是异常评论），本文选取了 YelpCHI 数据集[3]，这个数据集包括在芝加哥地区的 67395 个酒店和餐馆的评论信息（类似我们国家的美团评论，也属于电子商务的一种）。Yelp 网站提供滤波算法用于识别假/可疑的评论并将其分为过滤列表，过滤后的评论也网站上公开（虽然这个结果是不够准确的，但是仍然作为后来研究者的基准）。

我们在 http://odds.cs.stonybrook.edu 网站上下载的 YelpCHI 数据集，该网站公开提供大量的、不同领域的离群点检测数据集，我们选取 http://odds.cs.stonybrook.edu/yelpchi-dataset/的数据集下载。

下面简单介绍一下该数据集。该数据集有 67395 条数据，包括酒店和旅馆的评论信息，其中认为是虚假评论的评论占 13.23%，该数据集有 38063 个评论人，其中 20.33%认为是虚假评论发布者，该数据集有 201 个评价产品。

YelpCHI 数据库包括用户和产品之间的关系信息外，在评论数据集中存在多种元数据，包括评论、时间戳的文本内容，和星级评定。Yelp 有通用的滤波算法过滤出可疑评论，并在 Yelp 的业务页面显示推荐的评论。这个过滤的结果也作为我们检测虚假评论的基准，也就说我们的检测结果的评价是在该过滤结果的基础上做展示。

我们只选取了酒店的评论信息做虚假评论检测。其中标记为正常的评论有 53400 个，标记为异常的评论由 8141 个。

4.2 数据集预处理

本文下载好的数据库是经过数据清洗的，因此只需要从该数据集中提取合适的特征作为我们模型的输入，完成异常评论检测任务，这些特征要求尽量能够区分正常和异常的评论（判别性），或者说是能够表达异常或正常评论的特性。我们参考[3]文章，分别从评论者、评论产品和评论内容本身出发，提取所需特征，提取特征从行为和内容两个方面考虑。下面是我们选取的特征和相关介绍。

（1） 评论者和产品的特征（同时从两个角度出发）

行为	MNR	一天中产生的最大评论数量
	PR	正面评论的比例（评级 4-5 星）
	NR	负面评论的比例（评级 1-2 星）
	avgRD	第 i 个用户（产品）的评级的偏差avg($\|d_{i*}\|$),$\|d_{ij}\|$是第 i 个评级和第 j 个平均评级的评级偏差的绝对值：$\text{avg}_{e_{ij}\in E_{i*}}\|d_{ij}\|$，这里$d_{ij}=r_{ij}-\text{avg}_{e_{ij}\in E_{*j}}r(e)$
	WRD	加权评级偏差，其中评论按新旧程度加权:$\frac{\sum_{e_{ij}\in E_{i*}}\|d_{ij}\|w_{ij}}{\sum_{e_{ij}\in E_{i*}}w_{ij}}$,对于$w_{ij}=\frac{1}{(t_{ij})^{\alpha}}$($t_{ij}$表示评论$e_{ij}$的在评论 j 上的排序，$t_{ij}$=0.5 是衰减率)
	BST	突发，垃圾邮件通常是网站的短期成员，公式，$$x_{BST}(i)=\begin{cases}0 & ,if\ \ L(i)-F(i)>r\\ 1-\dfrac{L(i)-F(i)}{r} & ,otherwise\end{cases}$$ $L(i)-F(i)$是最后的评论和第一条评论间隔的天数
	ERD	用户（产品）评论的评分分布熵
	ETG	时间差熵的熵$\Delta_t s$，考虑到用户（产品）评论的时间排列，每个Δ_t表示连续对之间的时间间隔（新）
内容	RL	评论长度的单词数量（平均）
	ACS	内容相似性，用户（产品）评论之间的配对余弦相似度，评论用 bag-of-bigrams 表示
	MCS	最大值，内容相似性-所有评论对的最大的余弦距离

（2）评论的特征

行为	Rank	所有产品的评论的顺序
	RD	绝对评级偏离产品的平均评分[13]
	EXT	评级的末端：$x_{EXT}=1$，当评级为{4，5}时为 1，否则为 0
	DEV	评论e_{ij}的评级偏差的阈值 $$x_{DEV}(i)=\begin{cases}1,if\ \frac{|r_{ij}-avg_{e\in E_{*j}}r(e)|}{4}>\beta_1\\ 0,otherwise\end{cases}$$
	ETF	早期的时间框架 $$f(e_{ij})=\begin{cases}0,if T(i,j)-F(j)>\delta\\ \frac{T(i,j)-F(j)}{\delta},otherwise\end{cases}$$
	ISR	是否唯一用户
内容	PCW	全是大写字母的百分比
	PC	大写字母的百分比
	L	评论的长度
	PP1	第一人称代词的比例
	RES	含有！的感叹句比例
	DLu	基与 unigrams 的描述长度
	DLb	基于 bigrams 的描述长度
	F	审查频率，近似使用 LSH

这些特征作为每一条评论的表示，输入到我们的模型中将进行训练和测试。其中通过查找每条评论的评论者和产品的名称，去特征库中找到相应的评论者特征和产品特征，拼接到评论特征后面作为模型的输入，本文的网络输入的特征的维度是 33。

每个特征因为意义不同，因此值域也相差很大，这对于网络训练是有负面影响的，因为并不是数值大的特征重要性强，因此本文对每个特征用下式进行归一化：

$$F = \frac{f - \min(f)}{\max(f) - \min(f)},$$

F 表示归一化后的特征，f 表示原始特征，min 表示这个特征 f 中的最小值，max 表示最大值。

5 实验设置

应用于降维的自编码器由多个全连接层组成，接受每条评论的输入，产生的重构误差z_r，本文用欧氏距离和余弦距离两种距离表示（拼接到一起），同时和中间的隐变量z_c拼接到一起产生 3 维的输出（隐变量 1 维，2 个重构距离），所以自编码器的组成是：enc_Fc（33，20,tanh）-enc_Fc(20,10,tanh)-enc_Fc(10,1,none)-Zc-dec_Fc(1,10,tanh)-dec_FC(10,20,tanh)-dec_Fc(20,33,none)，估计网络的组成是 gmm_Fc(3,10,tanh)-Drop(0.5)-gmm_Fc(10,2,softmax)。其中，enc 表示编码层，Fc 表示全连接网络，括号里面的数字表示全连接层参数矩阵的维度，tanh 表示采用双曲正切激活函数，none 表示不使用激活函数，Drop 表示深度网络的一种学习技巧——Dropout。

正在确定网络组成后，本文每次选择批次大小为 1024 的训练数据输入到网络中，便于深度自编码网络和深度高斯混合模型网络的学习，本文的网络采用 adamoptimizer 自适应学习率和衰减系数的训练器，网络最后的输出打分同样经过归一化处理

$$S = \frac{s - \min(s)}{\max(s) - \min(s)},$$

然后选取 0.4 为阈值进行异常检测。

6 实验结果

数据集和网络配置分别在第 4 章和第 5 章介绍过了，本章主要介绍实验的结果。

6.1 评估方法

本文面对异常检测任务，采用准确率(precision rate)、召回率（recall rate）和综合评价指标（F-measure）三个评价指标进行评估：

准确率（P）=检索到的正确的异常评论数量/所有的正确的异常评论数量，

召回率（R）=检索到的正确的异常评论数量/检索到的异常评论数量。

准确率和召回率是互相影响的，理想情况下肯定是做到两者都高，但是一般情况下准确率高、召回率就低，召回率低、准确率高，如果做疾病监测、反垃圾，则是保准确率的条件下，提升召回率。

除了准确率和召回率外，综合评价指标是 Precision 和 Recall 加权调和平均

$$\mathrm{F}=\frac{(a^2+1)P*R}{a^2(P+R)},$$

本文选取 a=1，得到 F1-scores,

$$\mathrm{F}=\frac{2PR}{(P+R)}。$$

6.2 定量分析

原则上我们可以使用大量的正常样本做训练，然后只保留和异常样本一样多的正常样本和异常样本一起作为测试集，但是为了和传统的朴素贝叶斯方法和线性 svm 分类器方法检测异常进行比较，我们选取了的测试集合大小为 2443 个异常评论和 2443 个正常评论（因为朴素贝叶斯和 svm 分类器需要正负样本进行训练，因此不能使用所有的异常评论作为测试集），朴素贝叶斯分类器和 svm 分类器使用 5698 个正常评论和 5698 个异常评论作为训练集，本文的方法采用相同的大小 5698*2=11396 的正常评论作为训练集。朴素贝叶斯分类器和 svm 分类器的训练集和测试集如下表所示：

训练集		测试集	
正常评论	异常评论	正常评论	异常评论
5698	5698	2443	2443

本文方法的训练集和测试集如下表所示：

训练集		测试集	
正常评论	异常评论	正常评论	异常评论
11396	0	2443	2443

实验结果如下表所示：

	Precision	Recall	F1-scors
朴素贝叶斯	0.6390	0.6811	0.6594
线性 SVM	0.4437	0.5792	0.50255
本文的方法	**0.6892**	**0.7673**	**0.7262**

可以看出，本文的方法比起朴素贝叶斯和线性 svm 分类器具有不小的提升，不论是准确率还是召回率都是最高的，而且召回率的提升尤为明显，这对异常检测来说更为重要。

7 结论

本文面对电商产品评论的异常检测任务，提出了一种基于自编码的深度高斯混合模型聚类方法，在训练模型过程中只学习正常的评论特征，使用深度自编码模型对其进行降维，然后采用深度高斯混合模型对正常样本进行聚类，那么在测试时，距离所有聚类中心都比较远的点认为是离群点，即异常的评论数据，完成异常检测任务。比起传统方法，本文的方法可以使用更多的数据，不仅挖掘了更为普遍的内在规律，还满足了深度学习对大数据的需求，可以很好地完成异常评论检测。

参考文献

[1]Hinton G E, Salakhutdinov R R. Reducing the dimensionality of data with neural networks[J]. science, 2006, 313(5786): 504-507.

[2]Viroli C, McLachlan G J. Deep Gaussian mixture models[J]. Statistics and Computing, 2017: 1-9.

[3]Rayana S, Akoglu L. Collective opinion spam detection: Bridging review networks and metadata[C]//Proceedings of the 21th acm sigkdd international conference on knowledge discovery and data mining. ACM, 2015: 985-994.