软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2021,32(7):2204−2218 [doi: 10.13328/j.cnki.jos.006257]
©中国科学院软件研究所版权所有.
E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

基于变分自编码器的异构缺陷预测特征表示方法*

贾修一[1], 张文舟[1], 李伟漳[2], 黄志球[3]

[1](南京理工大学 计算机科学与工程学院,江苏 南京 210094)

[2](南京航空航天大学 航天学院,江苏 南京 210016)

[3](南京航空航天大学 计算机科学与技术学院,江苏 南京 211106)

通讯作者: 李伟漳, E-mail: liweiwei@nuaa.edu.cn

摘 要: 跨项目软件缺陷预测技术可以利用现有的已标注缺陷数据集对新的无标记项目进行预测,但需要两者之间具有相同的度量集合,难以用于实际开发.异构缺陷预测技术可以在具有异构度量集合的项目间进行缺陷预测,该技术引起了大量研究人员的关注.现有的异构缺陷预测技术利用朴素的或者传统机器学习方法为源项目和目标项目学习特征表示,所学习到的特征表示能力很弱且缺陷预测性能很差.鉴于深度神经网络强大的特征抽取和表示能力,基于变分自编码器技术提出了一种面向异构缺陷预测的特征表示方法.该模型结合了变分自编码器和最大均值差异距离,能够有效地学习源项目和目标项目的共性特征表示,基于该特征表示可以训练出有效的缺陷预测模型.在多组缺陷数据集上通过与传统跨项目缺陷预测方法及异构缺陷预测方法实验对比验证了所提方法的有效性.

关键词: 异构缺陷预测;变分自编码器;特征表示

中图法分类号: TP311

中文引用格式: 贾修一,张文舟,李伟漳,黄志球.基于变分自编码器的异构缺陷预测特征表示方法.软件学报,2021,32(7):2204−2218. http://www.jos.org.cn/1000-9825/6257.htm

英文引用格式: Jia XY, Zhang WZ, Li WW, Huang ZQ. Feature representation method for heterogeneous defect prediction based on variational autoencoders. Ruan Jian Xue Bao/Journal of Software, 2021,32(7):2204−2218 (in Chinese). http://www.jos.org.cn/1000-9825/6257.htm

Feature Representation Method for Heterogeneous Defect Prediction Based on Variational Autoencoders

JIA Xiu-Yi[1], ZHANG Wen-Zhou[1], LI Wei-Wei[2], HUANG Zhi-Qiu[3]

[1](School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China)

[2](College of Aerospace Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)

[3](College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China)

Abstract: Cross-project defect prediction technology can use the existing labeled defect data to predict new unlabeled data, but it needs to have the same metric features for two projects, which is difficult to be applied in actual development. Heterogeneous defect prediction can perform prediction without requiring the source and target project to have the same set of metrics and thus has attracted great interest. Existing heterogeneous defect prediction models use naive or traditional machine learning methods to learn feature representations

* 基金项目: 国家自然科学基金(61906090, U20B2064, 61773208); 江苏省自然科学基金(BK20191287, BK20170809); 中央高校基本科研业务费专项资金(30920021131); 中国博士后科学基金(2018M632304)

Foundation item: National Natural Science Foundation of China (61906090, U20B2064, 61773208); Natural Science Foundation of Jiangsu Province, China (BK20191287, BK20170809); Fundamental Research Funds for the Central Universities (30920021131); China Postdoctoral Science Foundation (2018M632304)

本文由“面向非确定性的软件质量保障方法与技术”专题特约编辑陈俊洁副教授、汤恩义副教授、何啸副教授以及马晓星教授推荐.

收稿时间: 2020-04-13; 修改时间: 2020-10-26; 采用时间: 2020-12-14; jos 在线出版时间: 2021-01-22

© 中国科学院软件研究所 http://www.jos.org.cn

between source and target projects, and perform prediction based on it. The feature representation learned by previous studies is weak, causing poor performance in predicting defect-prone instances. In view of the powerful feature extraction and representation capabilities of deep neural networks, this study proposes a feature representation method for heterogeneous defect prediction based on variational autoencoders. By combining the variational autoencoder and maximum mean discrepancy, this method can effectively learn the common feature representation of the source and target projects. Then, an effective defect prediction model can be trained based on it. The validity of the proposed method is verified by comparing it with traditional cross-project defect prediction methods and heterogeneous defect prediction methods on various datasets.

Key words: heterogeneous defect prediction; variational autoencoders; feature representation

软件缺陷预测技术是软件质量保证活动中非常重要的研究课题,基于机器学习方法进行缺陷预测可以从历史项目数据中学到软件度量和软件缺陷之间的联系,有效地帮助开发人员和测试人员在开发生命周期的早期预测模块(软件包、文件、类、函数或变更)中包含缺陷的可能性[1],从而降低软件缺陷带来的损失,节约宝贵且有限的资源.

现有的基于机器学习方法的软件缺陷预测模型需要首先设计衡量模块复杂度的度量并收集相关的缺陷数据集,之后才能在缺陷数据集上训练预测模型[2].大部分缺陷度量分为基于软件代码的软件度量和基于软件开发过程的软件度量,其中常用的基于软件代码的度量包括代码行数、Halstead 科学度量、McCabe 环路复杂度以及 CK 度量元,常用的基于软件开发过程的度量包括基于代码修改特征的度量和基于开发人员的度量.大多数软件缺陷预测模型的研究集中于项目间(within-project)软件缺陷预测[3−5],指的是模型的训练和预测都是基于同一个项目.然而对于新项目而言,其历史的缺陷数据是非常稀缺的,并且软件开发方法和语言的更新迭代十分迅速,如果仅仅只使用同一个项目的历史缺陷数据用于训练,由于训练数据的匮乏,往往很难构建有效且实用的缺陷预测模型以用于软件质量保障过程.为此,研究人员提出了跨项目(cross-project)软件缺陷预测[6−10],如图 1 左图所示,用其他项目的历史缺陷数据训练模型,在新的缺乏历史数据的项目上进行预测,从而可以充分地利用已有源项目缺陷数据,构建有效的目标项目缺陷预测模型.

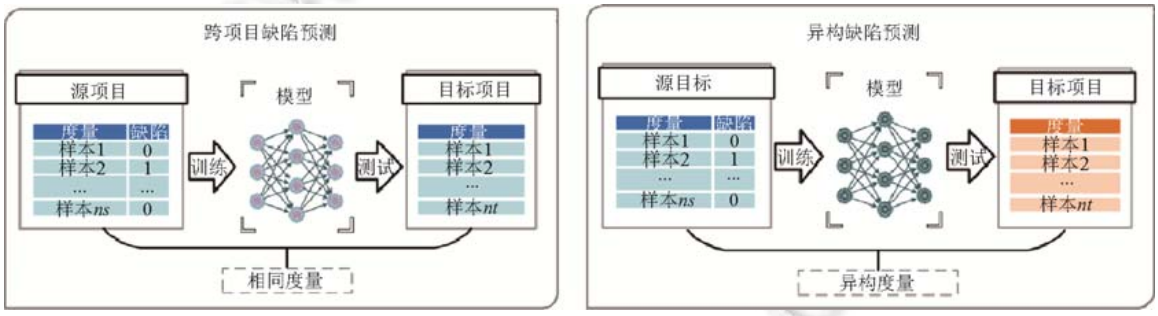


Fig.1 Cross-project software defect prediction and heterogeneous software defect prediction

图 1 跨项目软件缺陷预测和异构软件缺陷预测

大部分跨项目缺陷预测方法有一个严重的限制,即需要源项目和目标项目具有相同的软件度量[9],然而在现实情况下,大部分项目都具有异构的特征表示.例如,在 PROMISE 库中的大部分 NASA 数据集有 37 个度量元[5],AEEEM 数据集拥有 61 个度量元[8],它们之间唯一相同的度量是代码行数(LOC).由于公有度量的匮乏,现有的跨项目软件缺陷预测方法将难以适用,并且,大量有效的度量信息未被充分地利用.为此,研究人员提出异构缺陷预测(heterogeneous defect prediction)[10]来解决拥有不同度量集合的跨项目缺陷预测问题,即使项目之间拥有不同的度量集合,异构缺陷预测模型依旧可以在无法获取源代码的情况下利用已有的数据集学习度量分布和软件缺陷之间的映射关系.如图 1 右图所示,通过解决跨项目缺陷预测问题对同构特征的依赖,开发人员可以极大地降低收集缺陷数据带来的成本,为软件质量提供可靠的保障.

对于异构缺陷预测研究,该问题的主要难点有:(1) 源项目和目标项目的度量没有相同的语义,除了少数度

© 中国科学院软件研究所 http://www.jos.org.cn

量相同之外,大部分都没有任何对应关系.(2) 不同项目之间由于度量不同,其数据不但分布差别很大,而且特征表示结构也不尽相同,难以映射到相同的空间中进行学习.(3) 对于不同的项目数据集而言,由于开发过程千差万别,其缺陷的分布也各有千秋,难以用朴素的方法定量地衡量度量数值和缺陷分布之间的关系.

针对上述问题,本文提出了一种基于变分自编码器的特征迁移映射方法 T-VAE(transfer-variational autoencoder),可以将源项目与目标项目映射到一个共享的鲁棒隐式特征空间中,模型框架图如图 2 所示.首先,利用变分自编码器[11]对源项目和目标项目进行无监督预训练,可以将源项目与目标项目的度量元映射到同一特征空间.同时,由于变分自编码器从隐式空间采样解码引入的随机性,使得隐式的特征空间具有较强的鲁棒性.接着,通过引入最大均值差异约束源项目和目标项目间隐式特征分布的均值参数的距离,不但可以捕获到源项目与目标项目之间的共性特征,还能区分出两者之间的特性差异,分别由隐含层输出的均值向量和方差向量所对应.最后,通过加入判别网络带来的分类损失,可以有效地保证隐式特征空间拥有很好的线性可分能力,即使采用简单的逻辑回归也可以有效地预测出缺陷分布的趋势.此外,多组对比实验也验证了本文所提方法在异构特征跨项目缺陷预测问题上的有效性.

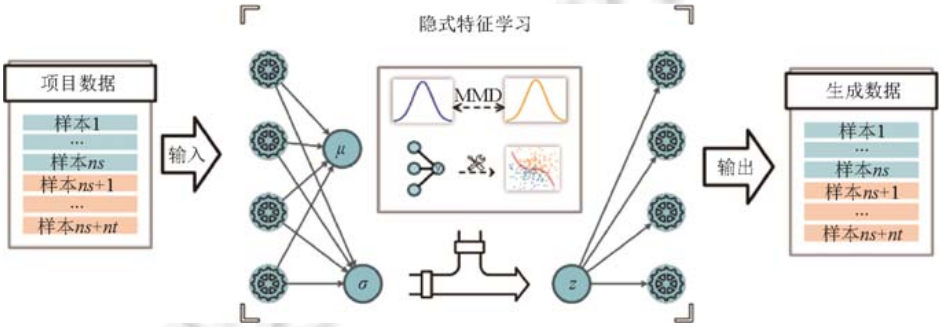


Fig.2 The framework of heterogeneous defect prediction model based on variational autoencoder

图 2 基于变分自编码器的异构缺陷预测模型框架图

本文第 1 节介绍异构缺陷预测的相关方法和研究现状.第 2 节介绍本文所需的基础知识,包括变分自编码器和最大均值差异.第 3 节介绍本文构建的基于变分自编码器的异构缺陷预测模型.第 4 节通过对比实验验证了所提模型的有效性.最后总结全文.

1 异构缺陷预测相关工作

现有的软件缺陷预测模型大多数都基于机器学习方法且集中于项目间软件缺陷预测.然而,对于新的项目而言,其历史缺陷数据是非常稀缺的,并且软件开发方法和语言的更新迭代十分迅速,如果仅只使用同一个项目的历史缺陷数据用于训练,往往很难构建有效且实用的缺陷预测模型以用于软件质量保障过程.研究人员提出了跨项目软件缺陷预测,大多数跨项目缺陷预测方法有一个重要的前提,即它们需要源项目和目标项目具有相同的软件度量,而摆脱了该限制的异构缺陷预测技术近年来引起了大量的研究兴趣.

异构缺陷预测是指使用从其他项目收集的异构度量数据来预测目标项目中软件实例的缺陷倾向性.它为缺陷预测提供了一个新的视角.最近,有相关工作提出了几种异构缺陷预测模型[10,12−14],用于预测具有异构度量的项目数据集中的缺陷(即源项目和目标项目具有不同的度量元集合).由于除通用度量外的其他度量可能具有良好的判别能力,Jing 等人[12]提出了一种异构缺陷预测方法,该方法利用了统一度量表示和基于典型相关分析的迁移学习技术.通过学习一对投影变换可以最大化源项目和目标项目之间的相关性,使目标项目的数据分布与源项目的数据分布相似.Nam 等人[10]提出了另一种用于异构缺陷预测的解决方案.他们首先采用度量选择技术来删除源项目中多余和不相关的度量.然后,根据度量相似度(例如分布或相关性)匹配源项目和目标项目的度量,构建度量标准集,进而预测目标项目中实例的标签.He 等人[13]提出了具有不平衡特征

© 中国科学院软件研究所 http://www.jos.org.cn

集的跨项目软件缺陷预测,以解决异构度量集问题.他们使用每个实例的分布特征向量作为新的度量标准,以实现缺陷预测.为了解决异构跨项目缺陷预测设置下的类别不平衡问题,Cheng 等人[14]提出了一种代价敏感迁移支持向量机方法.为了减轻不平衡数据的影响,他们通过将代价因素纳入支持向量机模型中,对有缺陷和无缺陷类别采用了不同的误分类成本.Zhang 等人[15]利用基于连接的聚类方法对跨项目缺陷预测问题进行研究,发现基于谱聚类的无监督方法比项目间缺陷预测的有监督模型更为有效.Li 等人[16,17]和 Tong 等人[18]都从核学习角度对异构缺陷预测中的类别不平衡问题和线性不可分问题进行了研究,并取得了非常好的效果.Gong 等人[19]在异构缺陷预测问题中引入神经网络模型进行分类,并通过最大均值差异距离来降低源项目和目标项目分布不匹配情况.Chen 等人[20]实验对比了多种异构缺陷预测方法,并得出 CTKCCA 性能最好的结论.

2 基础知识

本文所提方法主要基于变分自编码器和最大均值差异,下面就相关概念和基本知识予以介绍.

2.1 变分推断和变分自编码器

对于常见的缺陷度量数据,可以假设它们是由更高层的变量生成,并且这些隐变量满足特定的分布,一般代表着数据的内在结构或者某种抽象.例如,缺陷数据集可以看作是由度量代码的复杂程度、组织结构的混乱程度等隐式特征生成的数据.假设原始缺陷数据集为 $X=\{x_i\}_{i=1}^N$ 包含 N 个独立同分布的连续变量 x,这些数据是利用未观测到的隐变量 z 通过某些随机过程而生成.这个过程一般包含两个步骤.

(1) 从隐变量所服从分布 $p(z)$的概率密度函数中生成一个值 z_i;

(2) 根据值 z_i 条件概率分布 $p(x|z)$ 生成新的 x_i.

也就是说,原始数据集的边缘概率分布 $p(x)$可由对随机变量 z 积分 $p(x)=\int p(z)p(x|z)\mathrm{d}z$,其中,$p(z)$是隐变量服从的先验分布.

推断问题要解决的问题是,给定数据样本 x_i,如何推断出后验分布 $p(z|x_i)$.由贝叶斯公式可得后验分布计算如下:

$$p(z|x_i)=\frac{p(z)p(x_i|z)}{p(x_i)} \tag{1}$$

然而,$p(x|z)$ 的参数往往会非常复杂且难以计算,因而真实样本的边缘概率分布 $p(x)$是难以估计的,从而通过推断获取数据集的后验分布 $p(z|x_i)$ 以获取编码的分布往往也很困难.

解决上述问题常用的方法有蒙特卡洛马尔可夫链[21]和变分推断[22].使用蒙特卡洛 EM 算法进行优化时,每一步的采样极为耗时,难以处理大型的数据集.常用的变分推断的主要问题有,对数据的结构做了很强的假设或者只能基于近似寻找到次优的解[23],无法取得令人满意的效果.最近的相关工作利用神经网络的强大拟合能力结合反向传播算法,能够有效地解决推断问题,其中最受欢迎的框架之一就是变分自编码器.变分推断利用平均场变分推断来近似后验分布,变分自编码器最大的特点是利用神经网络同时拟合生成模型和推断模型.其中推断模型就是自编码器中的编码层,生成模型是自编码器中的解码层.

变分自编码的目标是通过最大化训练数据的边缘对数似然来学习参数化的隐式变量模型.它的边缘对数似然是由每个独立的数据点的边缘似然值求和所得,如下式:

$$\log p(x_1,\ldots,x_N)=\sum_{i=1}^{N}\log(x_i) \tag{2}$$

其中,$X=\{x_i\}_{i=1}^N$ 表示训练数据集.通过引入的近似后验分布 $q_\theta(z|x)$ 去拟合真实的后验分布 $p(z|x)$,那么单个样本的边缘似然可以写成下式:

$$\log p(x_i)=D_{KL}(q_\theta(z|x_i)\|p(z|x_i))+\mathcal{L}_{VAE}(\theta;x_i) \tag{3}$$

等式右手边第 1 项表示近似的后验分布与真实后验分布之间的 KL 散度,并且 KL 散度的值是恒大于 0 的.

© 中国科学院软件研究所 http://www.jos.org.cn

等式右手边第 2 项表示的是该数据点的边缘概率似然的变分下界(evidence lower bound,简称 ELBO),可以通过不断优化增加该下界的值,以不断用近似后验分布 $q_\theta(z|x)$ 来逼近真实的后验分布 $p(z|x)$.

式(3)可以通过重写负对数似然为下式:

$$\mathrm{E}_{\hat{p}(x)}[-\log p(x)]=\mathcal{L}_{VAE}(\theta;x)-\mathrm{E}_{\hat{p}(x)}[D_{KL}(q_\theta(z|x)\|p(z|x))] \tag{4}$$

其中,

$$\mathcal{L}_{VAE}(\theta;x)=\mathrm{E}_{\hat{p}(x)}[\mathrm{E}_{q_\theta(z|x)}[-\log q_\varphi(x|z)]]+\mathrm{E}_{\hat{p}(x)}[D_{KL}(q_\theta(z|x)\|p(z))] \tag{5}$$

并且, $\mathrm{E}_{\hat{p}(x)}[f(x)]=\frac{1}{N}\sum_{i=1}^{N}f(x)$ 函数 $f(x)$的期望.图 3 展示了该框架的主要思想,我们通过最小化式(5)中边缘负对数似然 $\mathcal{L}_{VAE}(\theta;x)$ 即可使我们估计的后验分布 $p(z|x)$ 不断逼近真实后验分布 $p(z|x)$.

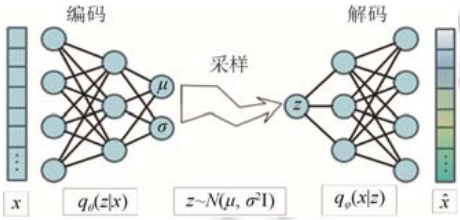


Fig.3 Variational autoencoder

图 3 变分自编码器

在式(5)中,等式右边的第 1 项表示经生成模型解码后数据与原始数据的差异,即重构误差,第 2 项表示模型所学的后验概率分布 $p(z|x)$ 与对隐式特征空间假设的先验分布 $p(z)$的差别.由此可见,变分自编码器所学的隐变量所服从的分布高度依赖于假设的先验分布.

在实际学习变分自编码器模型的过程中,隐式变量 z 会从推断模型中得到的分布 $q_\theta(z|x)$ 中进行采样,然后再将采样后的 z 输入至生成模型中进行解码,通过这个步骤,变分自编码器可以将隐式特征从单一重构损失最小所对应的概率最大值点转化为能以较高概率重构原始数据的某个概率分布,有效地提升了所学到的隐式特征的鲁棒性和抗噪声能力,也可以从一定程度上将源项目和目标项目的数据分布映射到同一个流形结构上去.

2.2 最大均值差异

在异构缺陷预测的研究问题中,虽然源项目与目标项目被映射到同一特征维度,但是由于两者特征的固有差异很大,无法保证学习到的隐式特征具有相同的语义,共享相似的分布.因此我们引入最大均值差异损失对源项目和目标项目隐式特征空间的后验分布进行约束,最大均值差异是通过计算两个分布的均值距离来度量两个分布的差异,如图 4 所示,通过最小化最大均值差异可以使两者的分布映射到同一空间之中.

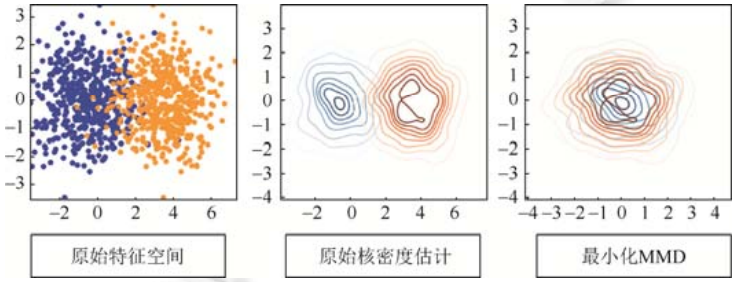


Fig.4 Learning distribution based on maximum mean discrepancy

图 4 利用最大均值差异学习分布

假设 $k: X\to X$ 是连续的,有界的半正定核,并且 H 是相应的再生希伯特核空间,并引入特征映射 $\Phi: X\to H$,那么分布 $p(x)$和 $p(y)$之间的最大均值差异(MMD)为

© 中国科学院软件研究所 http://www.jos.org.cn

$$\mathrm{MMD}(p(x),p(y))=\left\|\mathrm{E}_{x\sim p(x)}[\Phi(x)]-\mathrm{E}_{y\sim p(y)}[\Phi(y)]\right\|_H^2 \tag{6}$$

假设 X 和 H 同属于同一空间 R^d 且 $\Phi(x)=x$, 则最大均值差异减少的是分布 $p(x)$和 $p(y)$之间均值的差异,也即 $\mathrm{MMD}(p(x),p(y))=\left\|\mu_{p(x)}-\mu_{p(y)}\right\|_2^2$, 选择一个合适的映射Φ可以从一个更高阶的角度来衡量两个分布的差异.

3 基于变分自编码器的特征迁移映射方法 T-VAE

鉴于神经网络强大的表示能力和特征提取能力,本文提出一种基于变分自编码器的特征迁移映射方法,该方法由 3 部分组成.

首先,采用深度生成模型变分自编码器来提取源项目和目标项目之间潜在的公有特征分布.对于软件缺陷预测问题而言,一般采用的度量元表示的是软件模块的复杂度,同时,不同缺陷数据集的内在复杂度往往可以由少数隐式潜变量来表示.例如,对于两个面向不同功能或者用不同语言开发的应用程序,两者的数据分布差异一定非常大,然而,如果提取出更高层次语义的特征,如代码的规范程度、模块设计的耦合程度、可扩展性等度量,那么两者就处于相同的隐式语义空间,此时在一个项目上训练的模型就可以轻易地扩展到另外一个项目上.与此同时,如果有具体项目的特点信息,一个有效的生成模型就可以还原不同项目的原始分布.变分自编码器不仅能在复杂场景中进行推断,也可以有效地作为生成模型生成连续型数据,研究人员发现,它可以用于无监督的解耦表征学习中,并且学习到的隐式特征空间维度能够有效地对应数据中独立的属性.

其次,引入最大均值差异距离来度量源项目与目标项目在隐式特征空间中的距离,通过最小化该距离以提取出二者的共性隐式特征表示,同时也区分出各自的特性分布表示.虽然利用变分自编码器可以将源项目与目标项目映射到同一特征空间中,但是并不能保证源项目与目标项目的隐式特征具有相似的语义信息,甚至两者的分布差异依然很大.我们的最终目标是希望基于隐式特征表示的特征空间具有相似的条件概率分布.虽然变分自编码器可以约束隐式特征服从高斯分布,但在真实的应用场景中,为了保证重构误差尽可能地小,最终所学到的隐式特征分布之间的边缘概率差异依然很大,仅仅采用变分自编码器学习隐式特征表示并不能达成这个目标,需要通过最大均值差异距离进行度量.

最后,增加一层判别网络,用来评估学习到的隐式特征的分类性能.带有最大均值差异的变分自编码器虽然可以将源项目与目标项目的度量映射至相同分布的隐式空间,但是对于异构缺陷预测问题而言,最终的目的是要估计每个样本包含缺陷的概率,也就是条件概率 $p(y|x)$, 通过隐式特征空间的学习,我们所要估计的是在隐式特征空间下包含缺陷的条件概率 $p(y|z)$, 但是变分自编码器保留的是能够以最大概率重构原始样本数据集分布的信息,并不能一定保留那些可能使模块产生缺陷的相关因素,为了使最终学习到的隐式特征能够有效地为缺陷预测服务,我们在隐含层特征空间之后加入了一层判别网络,用来评估该隐式特征在源项目上的分类性能.通过反向传播误分类带来的损失,不仅可以学习到一个强大的非线性分类器,还能保留并提取原始数据集中与缺陷相关的信息,使模型有更好的分类效果.

3.1 利用变分自编码器学习软件缺陷数据的隐式特征

在以往的异构缺陷预测研究中,许多工作采用了统一度量表示[12],通过将不同语义的特征互相扩充补零来解决缺陷度量特征空间维度不一致问题,然而,这种方式会导致神经网络的容量增加,使得公共的语义特征信息难以被变分自编码器捕获.在本文工作中,我们直接对特征维度较小的数据集进行特征补零扩充,使其具有与特征维度高的数据集相同的特征数目,再将两者的样本数据输入到同一个变分自编码器网络中学习隐式特征分布.同时,我们假设隐含层输出编码 z 所服从的分布为正态分布,即 $p(z)=N(z;0,\mathbf{I})$.

不同于自编码器网络,变分自编码器网络在隐含层的输出包括两个维度μ和σ,分别表示编码 z 所服从的正态分布的参数均值和方差,即如下式:

$$q_\theta(z|x)=N(z;\mu,\sigma^2\mathbf{I}) \tag{7}$$

因此,后验分布 $q_\theta(z|x)$ 和先验分布 $p(z)$之间的 KL 散度可由下式计算,即为

© 中国科学院软件研究所 http://www.jos.org.cn

$$D_{KL}(q_\theta(z|x)\|p(z))=\int q_\theta(z)(\log p(z)-\log q_\theta(z))\mathrm{d}z=\frac{1}{2}\sum_{j=1}^{J}(1+\log((\sigma)^2)-(\mu_j)^2-(\sigma_j)^2) \tag{8}$$

然后从后验分布 $q_\theta(z|x)$ 进行采样得到新的隐含层编码 z,并将其输入至生成网络 $q_\varphi(z|x)$ 中生成新的 $\hat{x}$.

对于生成模型,可以使用伯努利分布去拟合原始的二值型数值,或者采用高斯分布拟合连续性的数值,在本节中,我们依旧采用神经网络去拟合该生成模型,并使用最小均方误差作为重构的损失,该部分的整体框架如 5 所示.

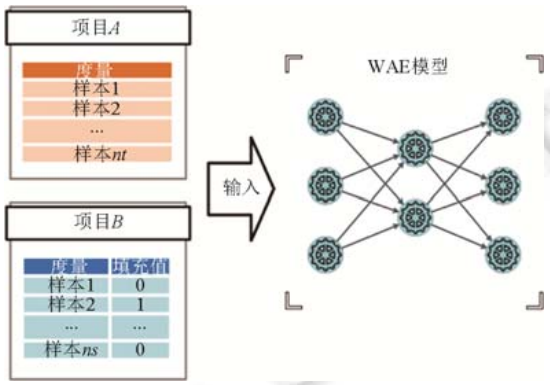


Fig.5 Heterogeneous feature mapping based on variational autoencoder
(source project and target project jointly train the network)

图 5 基于变分自编码器的异构特征映射(源项目与目标项目共同训练网络)

3.2 基于最大均值差异的项目个性与共性特征学习

为了更好地捕获源项目与目标项目之间的公共语义特征信息,我们引入了最大均值差异(maximum mean discrepancy)距离度量方式[24].边缘分布自适应方法最早由香港科技大学杨强教授团队提出[25],方法被命名为迁移成分分析(transfer component analysis),它能够很好地计算两个分布之间的差异.通过对源项目和目标项目的均值参数引入最大均值差异损失,可以有效地保证源项目与目标项目的后验概率分布 $q_\theta(z|x)$ 的均值参数拥有相似的分布,同时两者的差异通过方差参数体现.均值参数刻画了源项目与目标项目的共性特征,而方差特征凸显了源项目和目标项目的个性特征.

在训练变分自编码器时,为了能让网络同时学习源项目与目标项目的分布信息,每次输入的批量样本一半采样自源项目,一半采样自目标项目.为使两者在隐式空间享有相同的边缘概率分布,两者投影后的后验分布中均值向量的最大均值差异距离也被引入优化的目标.假设源项目为 $X^s=[x_1^s,x_2^s,\ldots,x_{n_s}^s]\in R^{d\times n_s}$,目标项目数据集为 $X^t=[x_1^t,x_2^t,\ldots,x_{n_t}^t]\in R^{d\times n_t}$,其中,$n_s$ 和 n_t 分别表示源项目与目标项目的样本个数,同时源项目经推断网络映射后的隐式特征 $z_s\sim N(\mu_s,\sigma_s^2\mathbf{I})$.目标项目的隐式特征 $z_t\sim N(\mu_t,\sigma_t^2\mathbf{I})$,且 $\mu=[\mu_s,\mu_t]$ 表示两者隐式特征分布的组合,那么源项目均值参数和目标项目均值参数之间边缘分布的最大均值差异可由下式定义:

$$\mathrm{MMD}(\mu_s,\mu_t)=\left\|\frac{1}{n_s}\sum_{i=1}^{n_s}\mu_i^s-\frac{1}{n_t}\sum_{i=1}^{n_t}\mu_i^t\right\|_2^2=\mathrm{tr}(\mu M\mu^T) \tag{9}$$

其中,M 是最大均值差异系数矩阵,假设 M_{ij} 表示 M 中的第 i 行、第 j 列元素值,那么 M_{ij} 可以通过下式计算得出:

$$M_{ij}=\begin{cases}\dfrac{1}{(n_s)^2}, & i\leqslant n_s, j\leqslant n_s\\ \dfrac{1}{(n_t)^2}, & i>n_s, j>n_s\\ -\dfrac{1}{n_s\times n_t}, & \text{otherwise}\end{cases} \tag{10}$$

© 中国科学院软件研究所 http://www.jos.org.cn

通过引入式(9)损失项,就可以保证源项目与目标项目经过映射后,隐式特征空间从均值相近的高斯分布中重采样.此时,隐变量所服从的方差参数描述了源项目和目标项目之间的真正差异所在,从生成模型可以有效地重构原始数据,还原原始的边缘概率分布.由此我们可以得出,隐式分布中的均值参数μ体现了源项目与目标项目之间公有的高层语义特征度量,而标准差参数σ体现了源项目和目标项目之间的差异,也就是项目各自的特性,从而可以有效地捕获更具价值的特征表示,以进一步指导软件开发中影响缺陷的宏观层面的相关因素,以帮助开发人员调整优化结构.

3.3 利用判别网络提取缺陷相关的高层特征表示

我们对推断模型输出的均值向量 μ_s 之后追加一层非线性感知机,用于预测在以该均值向量作为参数的后验概率分布 $q_\theta(z|x)$, 判断模块包含缺陷概率 $q_\theta(y|\mu)$ 的性能好坏,并不断优化其中的参数.由于我们只有源项目的标记信息,没有目标项目的标记信息,因此,基于源项目训练出来的分类模型 $q_\theta(y|\mu_s)$ 性能的好坏完全取决于:(1) 源项目的后验概率分布 $q_\theta(z_s|x_s)$ 与目标项目的后验概率分布 $q_\theta(z_t|x_t)$ 两者之间的匹配程度;(2) 源项目隐式特征表示的条件概率分布 $q_\theta(y|\mu_s)$ 与目标项目隐式特征表示的条件概率分布 $q_\theta(y|\mu_t)$ 两者之间的相似程度.在异构缺陷预测问题中,无论是源项目还是目标项目,提取缺陷度量时通常基于一个核心假设,即对于越复杂的软件模块而言,其包含缺陷的概率越大.因此,对于原始数据而言,如果能够很好地学习原始缺陷数据集的边缘概率分布 $p(X)$,我们就可以推断出它们之间的条件概率分布 $p(y|X)$ 也应该是相似的.

本文采用交叉熵计算判别模型预测出的包含缺陷的概率和真实是否有缺陷的标记作为损失,具体的损失项如下式所示:

$$H(y_s,\hat{y}_s)=-\sum_{i=1}^{n_s}y_s^i\log(\hat{y}_s^i) \tag{11}$$

其中,y_s 表示源项目数据是否包含缺陷的真实标记, $\hat{y}_s$ 表示判别网络对源项目数据预测是否包含缺陷的概率值.

结合上述各小节所述,基于公式(5)、公式(9)和公式(11),最终的模型目标函数如下:

$$\min_\theta \mathrm{E}_{\hat{p}(x)}[\mathrm{E}_{q_\theta(z|x)}[-\log q_\varphi(x|z)]]+\lambda_1\mathrm{E}_{\hat{p}(x)}[D_{KL}(q_\theta(z|x)\|p(z))]+\lambda_2\mathrm{MMD}(\mu_s,\mu_t)+\lambda_3H(y_s,\hat{y}_s) \tag{12}$$

首先,源项目数据集和目标项目数据集成对输入变分自编码器网络之中,通过推断网络得到隐式特征分布的参数μ和θ.然后,(1) 根据参数重采样得到新的隐式变量 z 并输入至生成网络,计算重构误差以及隐式分布和先验分布之间的 KL 散度,分别对应公式(12)目标函数的第 1 项和第 2 项;(2) 计算源项目对应的 μ_s 和目标项目对应的 μ_t 之间的最大均值差异距离,对应目标函数的第 3 项;(3) 计算以 μ_s 为判别网络输入得到的有缺陷概率和真实缺陷标记之间的交叉熵,对应目标函数第 4 项;最终逐步训练整个网络,网络架构基于 Tensorflow 2.0 框架实现,通过 Keras 提供的 ADAM(adaptive moment estimation)优化器来求解我们的目标函数. λ_1、λ_2 和 λ_3 为正则化参数.

4 实验分析

4.1 实验数据

我们在公开缺陷数据集上进行实验,包括 NASA、SOFTLAB、AEEEM 和 ReLink.在这些数据集中,有缺陷成分的百分比为 8.65%~50.52%.表 1 给出了数据集所对应的详细项目信息.

NASA 数据集广泛应用于缺陷预测[5],其静态代码度量包括大小、可读性、复杂性等,它们与软件质量密切相关.我们使用了 5 个项目,包括 CM1、MW1、PC1、PC3 和 PC4,因为这 5 个项目具有 37 个通用度量.

土耳其软件公司(SOFTLAB)由 AR1、AR3、AR4、AR5 和 AR6 项目组成,其中的项目是从 PROMISE 存储库中获得的,有 29 个度量元,其中包括 Halstead 和 McCabe 的圈数度量.

ReLink 中的数据集由 Wu 等人收集[26],通过增加缺陷数据的质量来提高缺陷预测性能.ReLink 中的缺陷信息已通过手动验证和纠正并由 3 个开源项目组成,每个项目都有 26 个复杂性度量.

AEEEM 由 D'Ambros 等人收集[3].每个 AEEEM 数据集包含 61 个度量标准:17 个源代码度量标准,5 个先前

© 中国科学院软件研究所 http://www.jos.org.cn

缺陷度量标准,5 个熵变化度量标准,17 个源代码熵度量标准和 17 个源代码波动度量标准.

Table 1 Experimental datasets

表 1 实验数据集

分组	项目	样本数量	度量数量	缺陷率(%)
NASA	CM1	327	37	12.84
	MW1	253		10.67
	PC1	705		8.65
	PC3	1 077		12.44
	PC4	1 458		12.21
SOFTLAB	AR1	121	29	7.44
	AR3	63		12.70
	AR4	107		18.69
	AR5	36		22.22
	AR6	101		14.85
ReLink	Apache	194	26	50.52
	Safe	56		39.81
	ZXing	399		29.57
AEEEM	EQ	324	61	39.81
	JDT	997		20.66
	LC	691		9.26
	ML	1 862		13.16
	PDE	1 497		13.96

4.2 评价指标及基准模型

在本文中,我们采用常用的评价指标 AUC 来评估缺陷预测模型的性能,该评价指标也广泛用于之前的研究工作[10,16,17].AUC 是受视工作特性曲线下方的面积,该曲线在二维空间中绘制,以假阳性率作为 x 坐标,真实阳性率(召回率)作为 y 坐标.AUC 由于不受类别不平衡的影响并且独立于预测阈值,因此被广泛使用,以用于评估不同模型的性能.

我们将所提方法 T-VAE 与项目间缺陷预测(WPDP)和跨项目缺陷预测(CPDP)方法(包括 TCA+[8]、VCB-SVM[27]和 ManuDown(M-Down)[28]以及异构缺陷预测方法 CCA+[12]、HDP-KS[10]、CTKCCA[16]和 SNN[19])进行了比较.

通过与项目间缺陷预测和跨项目缺陷预测方法进行比较可以提供异构缺陷预测方法(HDP)在实践中有效性的直接证据.其中,我们与基于逻辑回归的项目间缺陷预测方法(WPDP-LR)及基于神经网络的项目间缺陷预测方法(WPDP-NET)进行了比较.同时,我们还与方法 TCA+以及 VCB-SVM 进行了比较,它们都是传统跨项目缺陷预测中的方法,且都要求源项目和目标项目的实例具有完全相同的度量.M-Down 是一种基于排序的无监督预测模型,不需要任何标记信息甚至源项目.

另外,我们还与异构缺陷预测方法进行了比较:包括 CCA+、HDP-KS、CTKCCA 以及 SNN 这 4 种模型.这些方法在之前工作中有很好的预测性能.其中,CCA+使用 CCA 方法学习特征的映射,HDP-KS 采用了特征分布匹配策略,CTKCCA 主要集中于解决线性不可分以及类别不平衡问题,SNN 是一种基于最大均值差异的异构缺陷预测方法,同时也是利用神经网络模型来学习和预测.

基于上述预测设置和评价指标,我们使用 Wilcoxon 符号秩检验来验证两个模型在预测性能上是否有显著差异,当 p 值小于 0.05 时,就认为模型间的差异是不可忽视的.此外,我们同时使用 Cliff's δ来检查不同模型预测性能的差异在实际应用中是否重要[29].当 $|\delta|\leqslant 0.147$ 时,认为是可以忽略的(N);当 $0.147<|\delta|\leqslant 0.33$ 时,认为是小的(S);当 $0.33<|\delta|\leqslant 0.474$ 时,认为是中等(M);当 $|\delta|>0.474$ 时,认为是大的(L).如果一个模型取得正并且不可忽视的值($|\delta|>0.147$),则表示该模型相对于另一个模型更具有使用价值.

4.3 实验方法

我们使用来自 NASA、SOFTLAB、ReLink 和 AEEEM 的 18 个项目作为实验数据集,并执行异构跨项目缺陷预测.我们从 18 个项目中选择一个项目作为目标,然后依次使用其他组中的每个项目作为源项目.例如,当表 1 中的 NASA 组中的 CM1 充当目标时,存在 13(18−5)个跨项目预测组合.由于我们主要关注具有异构度量标准集

© 中国科学院软件研究所 http://www.jos.org.cn

合的数据集的预测,因此我们没有对数据集具有相同度量集的同一组中的项目进行缺陷预测.总共有来自 4 个小组的 18 个项目的 240 种可能的预测组合.

对于项目间缺陷预测问题,我们将数据集分为训练集和测试集并在缺陷预测模型中使用两折交叉验证[23,25].具体来说,我们将前半部分用于训练,后半部分用于测试,然后再以相反的方式将后半部分用于训练,前半部分用于测试.为了解决采样的随机性,我们随机拆分重复 50 次,则每个分类器都有 100 个测试结果.

在我们的实验中,模型由神经网络构成,实现方法基于 Python 以及 Tensorflow 2.0 框架,其中有多项超参数 λ_1、λ_2 和 λ_3,这些参数的搜索范围都为{1e–2,1e–1,1e0,1e+1,1e+2},在实验结果中汇报的值是多个源项目进行验证后的平均值.其中,每次验证的结果是在最优的参数配置下模型收敛时的平均性能.我们采用的策略是对于每个目标项目的同一种参数配置,以其能在所有的源项目取得最优平均值的参数组合作为最佳的参数配置(代码已发布在 https://github.com/NJUST-IDAM/T-VAE).

4.4 实验结果与分析

为了评估基于变分自编码器的异构缺陷预测方法 T-VAE 的有效性,我们研究了以下两个问题.

- RQ1:T-VAE 是否可以获得比项目间以及跨项目缺陷预测方法更好的结果?
- RQ2:T-VAE 是否比其他异构缺陷预测的方法更好?

RQ1: **T-VAE 是否可以获得比项目间以及跨项目缺陷预测方法更好的结果**?

为了验证这个问题的结果,我们将 T-VAE 模型与基于逻辑回归和单层神经网络的项目间缺陷预测以及跨项目缺陷预测方法 TCA+和 VCB-SVM 方法进行了对比实验,实验的结果见表 2.

Table 2 The comparison results between T-VAE and within-project/cross-project defect prediction methods

表 2 T-VAE 与项目间和跨项目缺陷预测方法性能比较

Target	WPDP-LR	WPDP-NET	TCA+	VCB-SVM	M-Down	T-VAE
CM1	0.653	0.466	0.642	0.614	0.624	**0.678**
MW1	0.612	0.432	0.685	0.579	0.712	**0.721**
PC1	0.787	0.649	0.694	0.648	**0.826**	0.743
PC3	0.794	0.670	0.627	0.587	**0.807**	0.736
PC4	**0.900**	0.865	0.687	0.583	0.757	0.814
AR1	0.582	0.481	0.624	0.660	0.527	**0.768**
AR3	0.574	0.537	0.733	0.660	0.825	**0.832**
AR4	0.657	0.573	0.751	0.667	**0.819**	0.798
AR5	0.804	0.497	0.819	0.846	**0.941**	0.890
AR6	0.654	0.522	0.582	0.598	0.563	**0.723**
Apache	0.714	0.655	0.697	0.698	**0.751**	0.745
Safe	0.706	0.651	0.721	0.704	**0.833**	0.775
ZXing	0.605	0.557	0.617	0.619	0.635	**0.635**
EQ	0.583	0.694	0.732	0.609	**0.793**	0.778
JDT	0.795	**0.818**	0.735	0.665	**0.785**	0.710
LC	0.575	0.653	0.633	0.639	0.671	**0.693**
ML	0.734	**0.743**	0.659	0.620	0.642	0.645
PDE	0.684	**0.696**	0.655	0.602	0.692	0.660
Mean	0.690	0.620	0.683	0.644	0.733	**0.741**
p-value	0.006	<0.001	<0.001	<0.001	0.711	–
Cliff's δ	0.358 (M)	0.608 (L)	0.503 (L)	0.747 (L)	0.006(N)	–

与项目间缺陷预测方法 WPDP-LR 和 WPDP-NET 相比,我们的方法有着巨大的优势,在绝大多数的数据集上都获得了最佳的效果(14/18),根据 Wilcoxon 符号秩检和 Cliff's δ检验的结果,T-VAE 相对于 WPDP-LR 有中等的优势,相对于 WPDP-NET 有着大规模尺度的优势.对于这一结果我们推测如下,大部分缺陷数据集中实例的个数都是比较少的(其中有 14/18 个项目的实例个数小于 1 000),实例个数过少直接导致有监督模型很难学习到泛化能力很强的分类模型,甚至还受类别不平衡问题以及过拟合问题的严重影响.在这种情况下,学习不同项目间的分布映射关系并拟合其他项目的缺陷分布情况反而能获得更好的泛化性能.其他相关的异构缺陷预测研究也有类似的结果,更充分表明了基于异构特征的跨项目软件缺陷具有很强的实用性.

与现有跨项目缺陷预测方法相比,根据 *p*-value 和 Cliff's δ的检验结果,T-VAE 相对于算法 TCA+和 VCB-

© 中国科学院软件研究所 http://www.jos.org.cn

SVM 具有显著的差异和大规模尺度的优势.T-VAE 和 M-Down 的性能之间并无显著区别,但需要注意以下几点.M-Down 是一种只利用了目标项目度量排序信息的无监督模型,其对度量元中缺陷分布信息的利用极其有限,且不引入额外假设时性能没有任何提升空间.T-VAE 所汇报的是在同一参数配置下,综合多个源项目取得最优平均值的结果,仅只需对源项目进行简单筛选并调节参数即可获得更优的性能.同时,T-VAE 还是一种表示学习方法,可以根据特定目标学习与缺陷更具相关性的特征表示,充分提取数据集度量元之间的信息.

综上,T-VAE 显著优于简单的项目间缺陷预测方法,并相对于现有的跨项目缺陷预测方法更具有实用价值.

RQ2: **T-VAE 是否比其他 HDP 的方法更好**?

为了回答该问题,我们将所提方法与现有的异构缺陷预测方法的性能进行了比较,其中,对比算法包括 HDP-KS、CCA+、CTKCCA 和 SNN,它们都是在与本文相同实现配置下进行的比较,实验结果见表 3.从结果可以看出,所对比的异构缺陷预测方法也有着令人印象深刻的性能,但总体而言,我们所提方法更优一些.从均值上看,对应于算法 HDP-KS、CCA+、CTKCCA、SNN,T-VAE 方法分别提升了 2.6%、42.7%、2.06%、6.16%.根据 Wilcoxon 符号秩检 T-VAE 与 CCA+、SNN 有着显著区别,其中相对于算法 HDP-KS、CCA+以及 SNN 都有着不可忽视的优势.算法 CTKCCA 与 T-VAE 有着相近的性能,但是我们的方法有着更高的平均得分,且在 10/18 个数据集上比它有更优的结果.同时,CTKCCA 利用核映射来处理线性不可分问题,其不仅需要专家知识设计有效的核函数,且难以处理大规模数据集.T-VAE 是一种基于神经网络的迁移学习方法,其不仅可以自动拟合复杂的数据分布,并且可以通过进一步增加数据规模以提升性能.同时,该方法还可以分析缺陷在隐式特征空间中的情况,提取两者之间的共性用于学习.

因此,综合而言,T-VAE 相对于大多数异构缺陷预测方法有较大优势,并更具有指导意义和实践价值.

Table 3 The comparison results between T-VAE and heterogenous defect prediction methods

表 3 T-VAE 与异构缺陷预测方法性能比较

Target	HDP-KS	CCA+	CTKCCA	SNN	T-VAE
CM1	0.689	0.480	**0.941**	0.641	0.678
MW1	0.709	0.532	**0.904**	0.692	0.721
PC1	0.728	0.499	**0.923**	0.668	0.743
PC3	0.728	0.502	**0.742**	0.678	0.736
PC4	0.690	0.511	0.680	0.677	**0.814**
AR1	0.687	0.518	0.683	0.686	**0.768**
AR3	**0.843**	0.518	0.543	0.805	0.832
AR4	0.782	0.514	0.683	0.756	**0.798**
AR5	**0.919**	0.565	0.496	0.843	0.890
AR6	0.660	0.503	0.651	0.659	**0.723**
Apache	0.748	0.639	**0.790**	0.695	0.745
Safe	0.775	0.496	0.596	0.722	0.775
ZXing	0.630	0.463	**0.781**	0.591	0.635
EQ	0.726	0.536	0.745	0.729	**0.778**
JDT	0.712	0.540	0.654	**0.726**	0.710
LC	0.659	0.512	**0.956**	0.720	0.693
ML	**0.650**	0.494	0.646	0.620	0.645
PDE	**0.668**	0.528	0.652	0.659	0.660
Mean	0.722	0.519	0.726	0.698	**0.741**
p-value	0.072	<0.001	0.711	<0.001	–
Cliff's δ	0.204 (S)	0.994 (L)	0.135(N)	0.386 (M)	–

4.5 参数影响分析

为了充分探讨在不同情况下,我们的算法受模型参数的影响状况,分别设置了两种场景用于分析不同参数权重对性能带来的影响.分别验证了在源项目数据规模较大和较小时,它对目标项目进行映射学习后的性能.具体而言,我们展示了分别以 PDE 和 AR6 为源项目时,在目标项目 PC4 上随着损失项各个参数的不同,其性能的变化趋势.

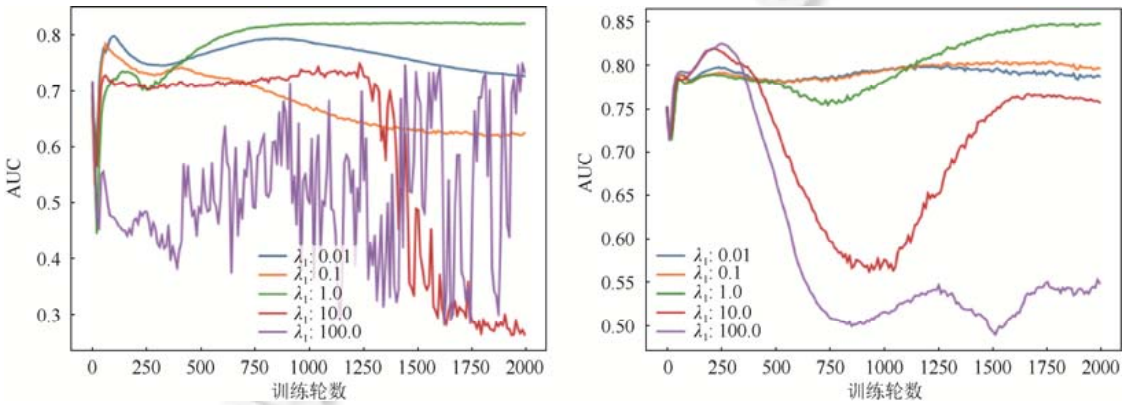
图 6~图 8 分别展示了不同损失项在不同权重下的性能表现.从图中可以看出,即使是相同的数据集,不同的损失权重也会导致有很大的差异,因此选择一个合适的参数对于获得有效的最终分类性能而言是非常重要的,

© 中国科学院软件研究所 http://www.jos.org.cn

例如,PDE⇒PC4 组合下,其λ_2在小于 1 时能收敛到较好性能,但在 AR6⇒PC4 组合下,其λ_2在大于 1 时可以收敛至较好性能,但也可以看出,较大范围的参数配置都可以得到较好的性能,进一步佐证了我们算法的鲁棒性.

对于不同的损失项,其参数调节的范围难以确定,许多情况下,过大的权重会导致模型的极度不稳定,例如,在 PDE⇒PC4 下,当后验分布与先验分布的 KL 散度以及分类损失交差熵的权重过大时,模型的性能不仅会很差,而且随着训练进程的推进,其预测的波动变化也很大.部分情况下会导致最终的性能走到相反的极端,这种情况可能是由于在对分布进行映射学习时,两者的条件概率分布映射到了相反的方向,由于目标项目的标记信息是完全未知的,因此出现这种情况也是有一定可能性的,通过将参数调整到更小的范围内可以有效地缓解该问题.

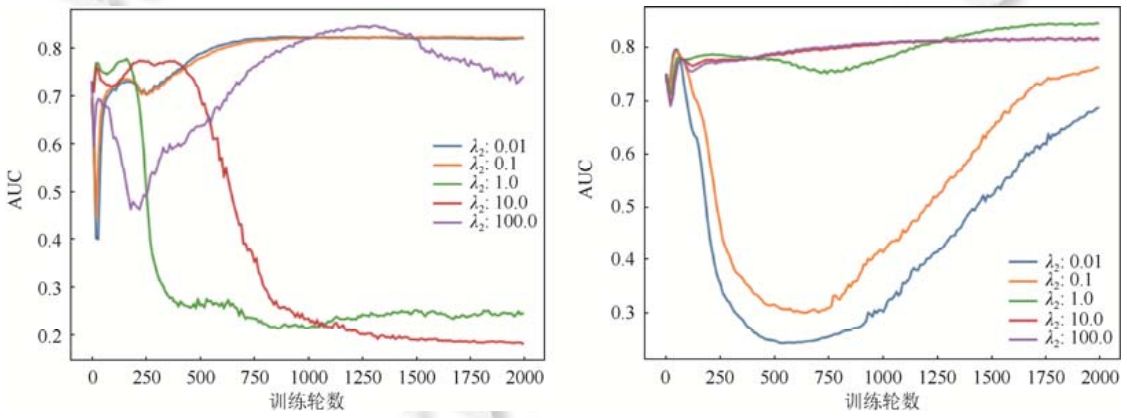
另外一个现象是,在许多场景中,初始训练时,在模型的参数处于完全随机的情况下,反而有可能得到较好的性能,随着训练过程不断推进,性能会有或多或少的下降,然后才会回升并趋于稳定.通过观察多组实验结果,这种情况是大量存在的,至于出现这种现象的原因,暂时尚未有合理的解释,还值得进一步研究与探索.



(a) PDE⇒PC4 (b) AR6⇒PC4

Fig.6 The performance variation of prior distribution difference under different weights based on given MMD and misclassification loss weight

图 6 在给定最大均值差异和分类损失权重情况下,先验分布差异在不同权重下的性能变化

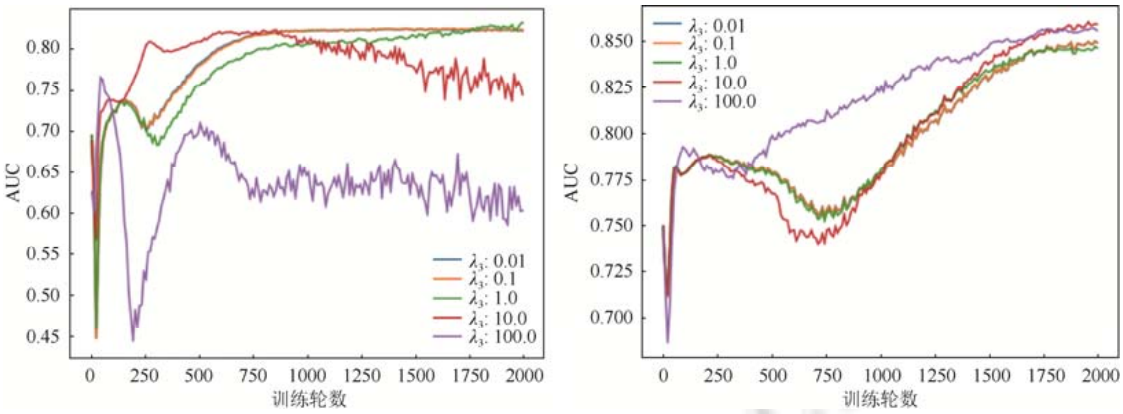


(a) PDE⇒PC4 (b) AR6⇒PC4

Fig.7 The performance variation of MMD under different weights based on given prior distribution difference and misclassification loss weight

图 7 在给定先验分布差异和分类损失权重情况下,最大均值差异在不同权重下的性能变化

© 中国科学院软件研究所 http://www.jos.org.cn



(a) PDE⇒PC4 (b) AR6⇒PC4

Fig.8 The performance variation of misclassification loss under different weights based on given prior distribution difference and MMD

图 8 在给定先验分布差异和最大均值差异权重情况下,分类损失在不同权重下的性能变化

总体而言,如果选择了合理的参数配置,就可以保证模型逐渐收敛至一个较好的性能,该模型可以有效地捕获源项目与目标项目的边缘概率分布并将其映射到新的子空间中,在该子空间中,这两个项目同时拥有相似的边缘概率分布的条件概率分布.

5 总 结

异构缺陷预测方法尝试解决异构特征之间的跨项目预测问题,该类方法具有非常强大的实用性.本文提出了一种基于变分自编码器的异构缺陷预测特征表示方法.针对现有异构缺陷预测方法,并不能很好地学习源项目与目标项目之间的隐式特征表示、拟合其中的分布信息等问题,本文基于变分自编码器,并结合最大均值差异对提取源项目与目标项目之间共性特征的方法进行了研究,通过进一步引入判别网络学习在隐式特征表示下的条件概率分布,可以有效地验证本文所提模型的特征抽象能力,通过在大量缺陷数据集的多组跨项目预测实验以及与多种缺陷预测方法比较,验证了本文所提出的异构缺陷预测方法不仅可以有效地学习两个项目之间的缺陷分布信息,还可以进一步提升缺陷预测的性能.

References:

[1] Hall T, Beecham S, Bowes D, Gray D, Counsell S. A systematic literature review on fault prediction performance in software engineering. IEEE Trans. on Software Engineering, 2011,38(6):1276–1304.

[2] Chen X, Gu Q, Liu WS, Liu WS, Liu SL, Ni C. Survey of static software defect prediction. Ruan Jian Xue Bao/Journal of Software, 2016,27(1):1–25 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4923.htm [doi: 10.13328/j.cnki.jos.004923]

[3] D'Ambros M, Lanza M, Robbes R. Evaluating defect prediction approaches: A benchmark and an extensive comparison. Empirical Software Engineering, 2012,17(4-5):531–577.

[4] Lee T, Nam J, Han D, Kim S. Developer micro interaction metrics for software defect prediction. IEEE Trans. on Software Engineering, 2016,42(11):1015–1035.

[5] Menzies T, Greenwald J, Frank A. Data mining static code attributes to learn defect predictors. IEEE Trans. on Software Engineering, 2006,33(1):2–13.

[6] Zimmermann T, Nagappan N, Gall H, Giger E, Murphy B. Cross-project defect prediction: A large scale experiment on data vs. domain vs. process. In: Proc. of the 7th Joint Meeting of the European Software Engineering Conf. and the ACM SIGSOFT Symp. on the Foundations of Software Engineering. New York: Association for Computing Machinery, 2009. 91–100.

© 中国科学院软件研究所 http://www.jos.org.cn

[7] He Z, Shu F, Yang Y, Li MS, Wang Q. An investigation on the feasibility of cross-project defect prediction. Automated Software Engineering, 2012,19(2):167–199.

[8] Nam J, Pan SJ, Kim S. Transfer defect learning. In: Proc. of the 35th Int'l Conf. on Software Engineering. New York: Association for Computing Machinery, 2013. 382–391.

[9] Ma Y, Luo GC, Zeng X, Chen AG. Transfer learning for cross-company software defect prediction. Information and Software Technology, 2012,54(3):248–256.

[10] Nam J, Fu W, Kim S, Menzies T, Tan L. Heterogeneous defect prediction. IEEE Trans. on Software Engineering, 2017,44(9): 874–896.

[11] Kingma DP, Welling M. Auto-encoding variational bayes. In: Proc. of the 2nd Int'l Conf. on Learning Representations. 2014.

[12] Jing XY, Wu F, Dong XW, Qi FM, Xu BW. Heterogeneous cross-company defect prediction by unified metric representation and CCA-based transfer learning. In: Proc. of the 10th Joint Meeting on Foundations of Software Engineering. New York: Association for Computing Machinery, 2015. 496–507.

[13] He P, Li B, Ma Y. Towards cross-project defect prediction with imbalanced feature sets. arXiv Preprint arXiv: 1411.4228, 2014.

[14] Cheng M, Wu GQ, Jiang M, Wan HY, You G, Yuan MT. Heterogeneous defect prediction via exploiting correlation subspace. In: Proc. of the 28th Int'l Conf. on Software Engineering and Knowledge Engineering. 2016. 171–176.

[15] Zhang F, Zheng Q, Zou Y, Hassan AE. Cross-project defect prediction using a connectivity-based unsupervised classifier. In: Proc. of the 38th Int'l Conf. on Software Engineering. New York: Association for Computing Machinery, 2016. 309–320.

[16] Li ZQ, Jing XY, Wu F, Zhu XK, Xu BW, Ying S. Cost-sensitive transfer kernel canonical correlation analysis for heterogeneous defect prediction. Automated Software Engineering, 2018,25(2):201–245.

[17] Li ZQ, Jing XY, Zhu XK, Zhang HY. Heterogeneous defect prediction through multiple kernel learning and ensemble learning. In: Proc. of the IEEE Int'l Conf. on Software Maintenance and Evolution. 2017. 91–102.

[18] Tong H, Liu B, Wang S. Kernel spectral embedding transfer ensemble for heterogeneous defect prediction. IEEE Trans. on Software Engineering, 2019. [doi: 10.1109/TSE.2019.2939303]

[19] Gong LN, Jiang SJ, Yu Q, Jiang L. Unsupervised deep domain adaptation for heterogeneous defect prediction. IEICE Trans. on Information and Systems, 2019,102(3):537–549.

[20] Chen HW, Jing XY, Li ZQ, Wu D, Peng Y, Huang ZG. An empirical study on heterogeneous defect prediction approaches. IEEE Trans. on Software Engineering, 2020. [doi: 10.1109/TSE.2020.2968520]

[21] Kass RE, Carlin BP, Gelman A, Neal RM. Markov chain Monte Carlo in practice: A roundtable discussion. The American Statistician, 1998,52(2):93–100.

[22] Blei DM, Kucukelbir A, McAuliffe JD. Variational inference: A review for statisticians. Journal of the American Statistical Association, 2017,112(518):859–877.

[23] Tschannen M, Bachem O, Lucic M. Recent advances in autoencoder-based representation learning. arXiv Preprint arXiv: 1812.05069, 2018.

[24] Quadrianto N, Petterson J, Smola AJ. Distribution matching for transduction. In: Advances in Neural Information Processing Systems. 2009. 1500–1508.

[25] Pan SJ, Tsang IW, Kwok JT, Yang Q. Domain adaptation via transfer component analysis. IEEE Trans. on Neural Networks, 2011, 22(2):199–210.

[26] Wu R, Zhang H, Kim S, Cheung SC. Relink: Recovering links between bugs and changes. In: Proc. of the 19th ACM SIGSOFT Symp. and the 13th European Conf. on Foundations of Software Engineering. New York: Association for Computing Machinery, 2011. 15–25.

[27] Ryu D, Choi O, Baik J. Value-cognitive boosting with a support vector machine for cross-project defect prediction. Empirical Software Engineering, 2016,21(1):43–71.

[28] Zhou YM, Yang YB, Lu HM, Chen L, Li YH, Zhao YY, Qian JY, Xu BW. How far we have progressed in the journey? An examination of cross-project defect prediction. ACM Trans. on Software Engineering and Methodology, 2018,27(1):1–51.

© 中国科学院软件研究所 http://www.jos.org.cn

[29] Romano J, Kromrey JD, Coraggio J, Skowronek J, Devine L. Exploring methods for evaluating group differences on the NSSE and other surveys: Are the *t*-test and Cohen'sd indices the most appropriate choices. In: Proc. of the Annual Meeting of the Southern Association for Institutional Research. Citeseer, 2006. 1–51.

附中文参考文献:

[2] 陈翔,顾庆,刘望舒,刘树龙,倪超.静态软件缺陷预测方法研究.软件学报,2016,27(1):1−25. http://www.jos.org.cn/1000-9825/4923.htm [doi: 10.13328/j.cnki.jos.004923]

贾修一(1983－),男,博士,副教授,博士生导师,CCF 高级会员,主要研究领域为机器学习,粒计算,数据挖掘.

李伟淖(1981－),女,博士,副研究员,CCF 专业会员,主要研究领域为机器学习,软件安全性.

张文舟(1994－),男,硕士生,主要研究领域为机器学习,软件缺陷预测.

黄志球(1965－),男,博士,教授,博士生导师,CCF 杰出会员,主要研究领域为软件工程,软件安全性,形式化方法.

© 中国科学院软件研究所 http://www.jos.org.cn