doi:10.3969/j.issn.1001-4616.2022.01.016

一种跨项目缺陷预测的源项目训练数据选择方法

盖金晶,郑　尚,于化龙,高　尚

(江苏科技大学计算机学院,江苏 镇江 212100)

[摘要]　跨项目软件缺陷预测(cross project defect prediction,CPDP)旨在实际的软件开发场景中,需要进行缺陷预测的目标项目可能是一个新启动项目,或这个项目已有的训练数据较为稀缺,需要利用其他项目已经搜集的训练数据来构建缺陷预测模型,其已经成为软件质量保证的一种手段,吸引了国内外研究人员的关注. 然而,面对不同的目标项目,训练数据的选择将直接影响预测模型的性能. 为了解决这个问题,本文描述了一种基于 JS 散度(Jensen-Shannon divergence)和相对密度的跨项目软件缺陷预测方法. 该方法首先通过将源项目和目标项目分别拟合高斯混合模型(Gaussian mixture model,GMM),再通过蒙特卡洛方法计算出目标项目和所有候选项目之间的 JS 散度. 其次,根据获得的 JS 散度选取与目标项目最接近的源项目;再次,提出相对密度概念,对选取的源项目训练数据进行有效选择. 最后,利用 CPDP 中常用分类器构建预测模型. 通过实验对比表明,本文方法不仅能够提高跨项目缺陷预测模型的性能,同时对不同分类器表现出较高的适应性.

[关键词]　跨项目,缺陷预测,软件质量,数据选择,JS 散度,相对密度,混合高斯模型,蒙特卡洛
[中图分类号]TP311.5　[文献标志码]A　[文章编号]1001-4616(2022)01-0110-08

A Cross Project Defect Prediction Method for Source Project Training Data Selection

Gai Jinjing,Zheng Shang,Yu Hualong,Gao Shang

(School of Computer,Jiangsu University of Science and Technology,Zhenjiang 212100,China)

Abstract:In real software development,a project which needs defect prediction is always new or without any historical data. It is necessary to use training data from several projects and performs prediction on another one. Therefore,cross project defect prediction(CPDP) has become a means of software quality assurance and been studied by researchers. However,the performance of prediction model can be directly affected by training data. In order to solve the problem,a cross-project software defect prediction method based on Jensen-Shannon divergence and relative density is proposed in this paper. Firstly,the Gaussian mixture model(GMM) is applied to the source and target projects respectively,and then the JS divergence between the target and all candidate projects is calculated by Monte Carlo method. Secondly,according to the obtained JS divergence,the source project that is most similar to the target project is selected. Thirdly,the concept of relative density is proposed to improve the training data quality of selected source project. Finally,some common classifiers are used to build the prediction model. Experimental results show that the proposed method can not only improve the performance of the prediction model,but also show high adaptability to different classifiers.

Key words: cross project, defect prediction, software quality, data selection, JS divergence, relative density, mixed Gaussian model,Monte Carlo

在软件开发的过程中,需求分析、软件设计以及开发人员的编码实现,每个过程都有可能由于开发人员的经验不足而出现软件缺陷. 这些软件缺陷可能会引起意想不到的问题,严重的情况下会给公司造成不可挽回的损失. 在软件开发的过程中不可避免会出现软件缺陷,而越早发现缺陷并且越早修复它们,则付出的代价越小. 若软件发布后,修复缺陷的成本将大大增加. 我们希望在软件开发初期就能够有效地检测出缺陷.

收稿日期:2020-07-27.
基金项目:江苏省自然科学面上基金项目(BK20191457)、江苏省高校面上基金项目(18JKB520011)、江苏省镇江市社会发展重点研发项目(SH2019021)、江苏科技大学高层次人才启动项目.
通讯作者:郑尚,博士,副教授,研究方向:智能软件工程. E-mail:szheng@just.edu.cn

跨项目软件缺陷预测(cross project defect prediction,简称 CPDP)旨在软件开发初期通过其他项目的历史数据,来构建软件缺陷预测模型,再在当前项目上进行预测,从而得到可能存在的潜在缺陷.

近年来,越来越多的研究者开始关注跨项目软件缺陷预测. 主要围绕同构跨项目缺陷预测(homogeneous cross-project defect learning)和异构跨项目缺陷预测(heterogeneous cross-project defect prediction)两方面. 其中,同构是指所有源项目和目标项目具有相同的度量元,而异构是指源项目与目标项目具有不同的度量元,根据具体情况,源项目之间也可能具有不同的度量元. 而本文则是针对同构跨项目缺陷预测开展研究.

Jureczko 和 Madeyski[1]从代码所有权(code ownership)角度,考虑了 3 种不同类型(工业界、开源界、学术界)的项目,对 CPDP 的可行性进行了分析. 他们的研究结果表明,跨项目进行软件缺陷预测的结果并不理想.

Turhan[2]为了深层次分析研究造成 CPDP 性能不理想的原因,引入了数据集漂移(dataset shift)的概念. 他们通过对数据集漂移类型进行分析,推荐了两类解决方法:基于实例的方法和基于分布的方法. Zimmermann 等[3]基于项目的上下文因素进行分析,共提出了 40 种不同的项目上下文因素来计算项目间的相似性. 他们的研究为源项目的选择提供了指引.

由于大量不相关的跨项目数据往往使建立高性能的预测模型变得困难. 为了克服这一挑战,许多研究人员专注于筛选与 CPDP 任务无关的源实例或特性.

Turhan 等[4]提出了 Burak 过滤法,他们通过计算训练集与目标项目中实例的欧式距离,选出最近的 K 个实例添加到训练集中. 他们的实验结果表明,Burak 过滤法能有效地提高 CPDP 模型的性能. He 等[5]提出了一种两阶段的筛选方法 TDS.第一阶段他们根据目标项目度量元的分布特征取值,通过欧式距离筛选出前 K 个候选源项目;第二阶段从前 K 个候选源项目中通过 Burak 过滤法或 Peters 过滤法[6]选出与目标项目最为接近的实例. 李勇等[7]随后提出了相似的两阶段筛选方法,他们根据余弦距离选出与目标项目最为相关的前 K 个候选源项目,随后借助 Peters 过滤法进一步选出相关实例.

除了上述基于实例选择的研究成果,国内外学者也对基于分布的跨项目缺陷预测开展了相关研究. Nam 等[8]提出一种迁移学习方法,迁移学习通过调整源项目,使得源项目与目标项目的特征相似. 并且他们进一步拓展 TCA,得到 TCA+来提升跨项目缺陷预测性能.

Zhao 等[9]提出了 NN 过滤器,他们通过删除那些不在目标项目数据的最近邻居中出现的源项目实例,这样使得源项目和目标项目在分布上更加相似. Ma 等[10]提出了一种名为 transfer naive Bayes(TNB)的方法,该方法首先通过 data gravitation(DG)方法[11]对源实例进行权重调整,以削弱不相关源数据的影响,然后对这些权重调整后的源数据构建朴素 Bayes 分类器.

最近,一些研究表明,在目标项目中添加一定比例的标记数据,可能有助于提高 CPDP 的性能. Chen 等[12]提出了一种新的方法(DTB),它在目标项目中使用少量的标记数据,同样使用 DG 方法初始化源项目数据的权值,然后利用 TrAdaboost[13]建立预测模型,利用目标项目中有限数量的标签数据对源数据进行重估.

上述跨项目缺陷预测方法虽已经取得了一定的成果,但是仍有改进的空间. 具体来说,即基于相似性选取的方法仍不精确,同时盲目使用大量的训练数据所训练出的预测模型容易导致较高的误报率,因此,本文在一对一同构跨项目缺陷预测领域做了相应的改进工作:

(1)从数据分布的角度,利用 JS 散度测算找到与目标项目最相似的源项目,提升源项目选取的准确度;

(2)提出相对密度,对选定的源项目进行数据选择,提高预测模型的精度;

(3)通过实验验证,本文方法对不同项目的训练集和分类器的选择,皆表现出较好的性能.

1 相关概念

1.1 JS 散度

JS 散度也称 JS 距离,是 KL 散度的一种变形. Kullback-Leibler divergence(KL 散度[14])又称为相对熵,信息散度,信息增益. KL 散度是两个概率分布 P 和 Q 差别的非对称性的度量. 典型情况下,P 表示数据的真实分布,Q 表示数据的理论分布、模型分布、或 P 的近似分布. 那么离散变量 KL 散度见式(1):

$$KL(P||Q)=\sum_{x\in D}P(x)\log\frac{P(x)}{Q(x)},\tag{1}$$

式中,$P(x)>0$,$Q(x)>0$,且规定 $\log\frac{P(x)}{P(x)}=0$.

由于 KL 散度不具有对称性,即 $KL(P\|Q)\neq KL(Q\|P)$,为了解决这个问题,有人提出了 Jensen-Shannon divergence(JS 散度)[15] 作为相似度度量的指标. 现有两个概率分布 P 和 Q,其 JS 散度公式如式(2):

$$JS(P\|Q)=\frac{1}{2}KL\left(P\|\frac{P+Q}{2}\right)+\frac{1}{2}KL\left(Q\|\frac{P+Q}{2}\right). \tag{2}$$

在计算 JS 散度的过程中使用了蒙特卡洛[16] 方法(Monte Carlo),蒙特卡洛法是一种用来模拟随机现象的数学方法,这种方法在模拟中能直接反映过程中的随机性. 如公式(3)所示:

$$D_{MC}(P\|Q)=\frac{1}{n}\sum_{i=1}^{n}\log p(x_i)/q(x_i)\rightarrow D(P\|Q). \tag{3}$$

不同于 KL 散度,JS 散度的值域范围是[0,1],相同为 0,相反则为 1. 相比较于 KL,对相似度的判别更准确了. 同时其对称性能让散度度量更准确. 因此,本研究拟利用 JS 散度去衡量任意两个项目的相似性,即 JS 散度越接近,两个项目的分布也就越相似.

1.2　相对密度

经过 JS 散度选定与目标项目分布最为相似的源项目需要对其训练数据选择,以便构建预测模型. 一般来说,若能精确地测出每一个训练样本的概率密度,从噪声点和离群点中区分出具有重要信息的样本则变得比较容易. 但是,在高维特征空间中,获得精确的概率密度是极为困难的,要获取相对精确的概率密度也是十分耗时的. 因此,本文采用了一种避免概率密度测量的方法,即精确提取任意两个训练样本的概率密度的比例关系,我们称这种反映比例关系的信息为相对密度.

为了计算相对密度,本文使用了与基于 K 最近邻概率密度估计法(K-nearest neighbors-based probability density estimation,KNN-PDE[17])相似的策略. 作为一种非参数概率密度估计方法,KNN-PDE 通过测量每个训练实例的 K 最近邻距离来估计多维连续空间中的概率密度分布. 当训练实例的数量达到无穷大时,从 KNN-PDE 获得的结果可以近似收敛到实际概率密度分布. 因此,本文提出的相对密度策略也采用了 K 最近邻距离估计相对密度.

假设一个数据集有 N 个样本,对于每个样例 x_i,找到它的第 K 个最近邻并计算它们之间的距离 d_i^K. 其中,d_i^K 越大,样本 x_i 的密度就越低. 同时,我们知道无论噪声或者离群值都应该出现在低密度区域,因此,d_i^K 可以用作评估每个实例重要性的方法. 但是,要为高密度实例提供较大的值,而为那些低密度实例提供较低的值,我们应将 d_i^K 转换为倒数 $1/d_i^K$. 这里,我们就将一个样本的 K 最近邻距离的倒数称之为其相对密度. 不难看出,任意两个样本之间的相对密度和它们之间 K 近邻距离的比例关系成反比. 如公式(4)所示:

$$\frac{1/d_i^K}{1/d_j^K}=d_j^K/d_i^K. \tag{4}$$

显然,在相对密度方法中,K 值的选择是十分重要的. 若 K 值太小,则很难将低样本从普通样本中区分出来;若 K 值太大,则那些重要样本与低样本的区别便会变得模糊不清,而这些微小的差距将更难获取. 因此,需要给参数 K 选择合适的值.

2　研究方法

文章以源项目数据选择为研究对象,从项目分布的相似性角度出发,首先利用 JS 散度进行源项目选择,其次提出相对密度进行训练数据选择,最后采用 CPDP 中常见的分类器对数据进行训练,并将模型用于目标项目的预测. 具体的研究框架和方法描述分别见图 1 和表 1.

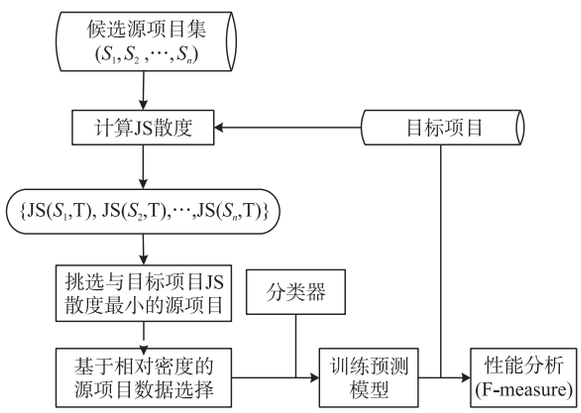

图 1　研究框架

Fig. 1　Research framework

2.1　数据标准化

本文使用 Z-Score 标准化方法来处理数据. Z-Score 标准化方法是常见的数据处理方法,通过将不同量级的数据转化为同一量级 Z-Score 分值进行比较.

该方法计算出原始数据的均值μ和标准差σ,对数据进行标准化处理. Z-Score 标准化方法的公式可表示如式(5):

$$Z=\frac{x-\mu}{\sigma}. \tag{5}$$

2.2 基于 JS 散度的源项目确定

本文在对候选源项目集使用 Z-Score 标准化处理之后,将进行 JS 散度的计算,进而完成源项目的确定.

根据前文描述,JS 散度能够度量两个项目数据分布的相似度,且 JS 越小证明分布越接近. 因此,本节工作主要是计算所有源项目与目标项目的 JS 散度,并选取与目标项目最接近的源项目.

表 1 研究方法

Table 1 Research method

方法描述:一种跨项目缺陷预测的源项目训练数据选择方法
输入:所有源项目$\{S_1,S_2,\cdots,S_n\}$和目标项目T
输出:F-measure
步骤 1:计算出所有源项目和目标项目之间的 JS 散度$\{JS(S_1,T),JS(S_2,T),\cdots,JS(S_n,T)\}$
步骤 2:挑选步骤 1 中计算得到最小的 JS 散度,从而确定源项目S_i
步骤 3:计算选定源项目的每个样本的相对密度,并进行重要性排序
步骤 4:设定阈值进行训练数据选取
步骤 5:采用 CPDP 中常用的分类器对选取的数据进行训练
步骤 6:将得到的预测模型应用到目标项目进行性能验证,获得相应的 F-measure

具体流程如表 2 所示. 首先,将候选源项目集和目标项目分别拟合混合高斯(Gaussian mixture model, GMM)[18]模型. 高斯混合模型可以看作是由K个单高斯模型组合而成的模型,这K个子模型是混合模型的隐变量(hidden variable). 一般来说,一个混合模型可以使用任何概率分布,这里使用高斯混合模型是因为高斯分布具备很好的数学性质以及良好的计算性能. 其次通过蒙特卡洛公式,计算候选项目集和目标项目的 JS 散度,其中$JS(S_i,T)$表示为当前第i个候选项目与目标项目计算得出的 JS 散度值;并通过比较所得 JS 散度的大小,选取出与目标项目 JS 散度最小的源项目,即确定其为训练项目.

表 2 源项目挑选流程

Table 2 Source project selection

$GMM=\{GMM_{S1},GMM_{S2},\cdots,GMM_{Sn},GMM_T\}$
步骤 2:根据公式(1)(2)(3),计算所有拟合 GMM 模型的 JS 散度 $JS=\{JS(S_1,T),JS(S_2,T),\cdots,JS(S_n,T)\}$
步骤 3:挑选与目标项目的 JS 散度最小的值,即是与目标项目分布最相似的源项目,将其输出,记为S_i

2.3 基于相对密度的源项目数据选择

挑选后的源项目仍包含少量的噪声或离群点样本,如若盲目选择将影响预测的效果. 为了筛选合适的训练数据,本文通过相对密度反映类别中每个实例的重要性,并有选择地截取数据,提高训练模型的精度. 具体的方法流程如表 3 所示.

首先,根据 2.2 选取的源项目,统计其所有样本数量,记为N;其次,计算每个样本与第K个最近邻的距离,获得相对密度;最后,根据相对密度进行重要性排序,同时设定阈值p(percent)选择合适数量的样本作为训练集. 由表 3 可以看出,参数K和p的设定将决定模型训练的质量,文中将在后续章节讨论两个参数的选择.

表 3 源项目的训练数据选择

Table 3 Training data selection of the source project

基于相对密度的源项目数据选择流程
输入:源项目数据集S_i
输出:筛选过的数据集$S_{i\,new}$
步骤 1:统计S_i的样本个数,记为N
步骤 2:计算每个样本到其第K个最近邻的距离,获得相对密度,并根据其值大小排序,确定每个样本实例的重要性
步骤 3:根据排序的样本,设定训练集选取的p(percent),其中p为总体样本的百分比
步骤 4:输出筛选过的新源项目数据$S_{i\,new}$

2.4 构建跨项目软件缺陷预测模型

为了证明我们方法的适应性,我们分别采取 CPDP 中常用的分类器逻辑回归(LR)、贝叶斯(NB)、支持向量机(SVM)、K近邻(KNN)训练预测模型,并用于目标项目加以验证,对得到的 CPDP 性能进行对比分析.

3　实验验证

3.1　数据集与评价指标

3.1.1　数据集

文中的数据集是来自 Jureczko 和 Madeyski[19] 收集的 PROMISE 库中的开源项目,其已经被广泛应用于跨项目缺陷预测研究. 表 4 展示了数据集的基本信息,包括项目名称、项目版本、实例数量和缺陷率.

表 4　PROMISE 数据集

Table 4　PROMISE data sets

项目名称	项目版本	实例数	缺陷比例%	项目名称	项目版本	实例数	缺陷比例%
ant	1.7	745	22.3	prop	6	660	10.0
camel	1.6	965	19.5	tomcat	6.0	858	9.0
jedit	4.3	492	2.2	velocity	1.6	229	34.1
lucene	2.4	340	59.7	xerces	1.4	588	74.3
poi	3.0	1077	63.6				

3.1.2　评价指标

在软件缺陷预测中,合理的评价指标能更好地评估预测结果,下文将介绍软件缺陷预测中的基本的评价指标.

软件缺陷可以看做是二分类问题,若将有缺陷的模块设置为正例,无缺陷模块为反例,则每个实例的分类过程中可能会出现以下 4 种情况:实际为有缺陷类被正确分类为有缺陷类,即真正例(true positive,TP);实际为无缺陷类被错误分类为有缺陷类,即假正例(false positive,FP);实际为无缺陷类被正确分类为无缺陷类,即真反例(true negative,TN);实际为有缺陷类被错误分类为无缺陷类,即假反例(false negative,FN). 在人工智能中,混淆矩阵是表示精度评价的一种标准格式,用 n 行 n 列的矩阵形式表示,如表 5 所示.

表 5　混淆矩阵

Table 5　Confusion matrix

真实情况	预测结果	
	有缺陷模块	无缺陷模块
有缺陷模块	真正例(TP)	假反例(FN)
无缺陷模块	假正例(FP)	真反例(TN)

根据上述描述,文章将采用 F-measure 作为本研究的评价指标,具体描述如下:

Precision:正确分类为正样本的实例数目与分类为正样本的实例数目的比率.

$$Precision=\frac{TP}{TP+FP}. \tag{6}$$

Recall:正确分类为正样本的实例数目与所有正样本实例数目的比率.

$$Recall=\frac{TP}{TP+FN}. \tag{7}$$

F-measure:对精确度和召回率的综合衡量. f 值越高,表现越好.

$$F\text{-}measure=\frac{2\times Precision\times Recall}{Precision+Recall}. \tag{8}$$

3.2　实验设置与实验结果

本文为了确认本方法是否优于其他方法,我们在 PROMISE 数据集上与主流的跨项目缺陷预测方法以及设定的基线方法进行了对比.

首先,对 8 种方法进行简述,如下所示:

(1)DTB[12] 方法:在目标项目中使用少量的标记数据,用来提高 CPDP 的效果.

(2)TrAdaboost[13] 方法:构建预测模型,利用目标项目中有限数量的标签数据对源数据进行重估.

(3)TNB[10] 方法:通过 data gravitation(DG)方法对源数据进行权重调整,然后用权重调整后的源数据构建朴素贝叶斯分类器.

(4)TCA+[8] 方法:转移成分分析方法,仅使用跨项目数据进行预测.

(5)NN filter[9] 方法:通过删除非目标项目最近邻的源项目数据来对源项目数据进行筛选.

(6)KMM[20] 方法:是 KMM-MCW 中的主要步骤,使用 MMD 最小化来对齐分布,降低域间差异,跳过概率密度估计,直接根据样本估计权重.

(7)KMM-MCW[21]方法:多分量权重(MCWs)学习模型来分析源项目中多个分量从而不断优化.

(8)基线方法:不做任何数据选择,直接对使用源项目训练模型进行目标项目预测.

其中,TrAdaboost、TCA+、KMM 和 KMM-MCW 的源代码由对应文章提供网站下载. DTB、TNB 和 NN 过滤器则是根据文章方法重新复现. 值得注意的是,方法(1)~(7)皆选取逻辑回归作为分类器,为了公平比较,本文方法和基线方法也选择逻辑回归作为分类器. 同时,文章在计算 JS 散度的过程中,设定拟合的混合高斯模型数量为 5. 且在源项目数据选择过程中,$K=\{\lceil\sqrt{N}/4\rceil,\lceil\sqrt{N}/2\rceil,\lceil\sqrt{N}\rceil,\lceil 2\sqrt{N}\rceil,\lceil 4\sqrt{N}\rceil\}$,$p$ 以样本数的 10%依次递增,为了获得最优参数,采用了网格搜索(Grid Search)方法来确定最优的 K 和 p 参数组合. 对比实验结果如表 6 所示.

表 6　各类方法的 F-measure 结果比较

Table 6　F-measure comparison of various methods

数据集	本文方法	KMM-MCW	DTB	TrAdaboost	TNB	TCA	NN	KMM	Baseline
ant	0.34	0.421	0.394	0.396	0.407	0.394	0.37	0.413	0.345
camel	0.377	0.284	0.295	0.304	0.26	0.263	0.26	0.288	0.242
jedit	0.036	0.072	0.043	0.048	0.044	0.049	0.05	0.057	0.087
lucene	0.648	0.645	0.6	0.647	0.625	0.571	0.597	0.597	0.389
poi	0.662	0.726	0.753	0.754	0.738	0.639	0.679	0.72	0.403
synapsel	0.536	0.465	0.462	0.465	0.388	0.451	0.458	0.448	0.398
tomcat	0.217	0.213	0.208	0.205	0.188	0.198	0.18	0.196	0.203
velocity	0.547	0.452	0.438	0.438	0.39	0.427	0.414	0.435	0.343
xerces	0.823	0.802	0.805	0.811	0.86	0.583	0.666	0.786	0.432
均值	0.465	0.453	0.444	0.452	0.433	0.397	0.408	0.438	0.316

除对比方法实验之外,本文为验证所提方法能够提高不同分类器构建的模型性能,在逻辑回归(LR)、贝叶斯(NB)、支持向量机(SVM)、K-近邻(KNN)这几种 CPDP 中常见分类器上也做了对比的实验,具体结果如表 7 所示.

表 7　不同分类器下本方法获得的 F-measure 结果比较

Table 7　F-measure comparison under different classifiers

数据集	本文方法-KNN	KNN	本文方法-NB	NB	本文方法-SVM	SVM
ant	0.36	0.31	0.326	0.477	0.331	0.422
camel	0.196	0.249	0.378	0.295	0.36	0.304
jedit	0.052	0.046	0.033	0.074	0.026	0.042
lucene	0.16	0.403	0.695	0.48	0.606	0.576
poi	0.436	0.392	0.814	0.405	0.785	0.601
synapsel	0.521	0.384	0.485	0.487	0.522	0.56
tomcat	0.222	0.223	0.247	0.33	0.24	0.264
velocity	0.514	0.364	0.545	0.36	0.534	0.461
xerces	0.794	0.432	0.789	0.436	0.787	0.601
均值	0.362	0.312	0.479	0.372	0.466	0.426

根据表 6 结果可以发现,与其他方法相比,本文方法最终选取的训练数据在大多数项目上取得较高的 F-measure,且整体结果的均值高于其他方法. 实验结果表明,本文方法能够在最大程度地利用源项目情况下,根据 JS 散度和相对密度选择合适的训练集构建预测模型,且获得较优的性能.

从表 7 可以发现,不同分类器结合本文方法在整体结果的均值都有了提升,且在大多数项目提高预测性能. 实验结果表明,本文方法能够适应于 CPDP 中常用的分类器,并提高模型性能.

4　参数讨论

方法中所涉及的参数 K 和 p(percent). 本文主要根据样本的数量,K 值的选择范围为:$K=\{\lceil\sqrt{N}/4\rceil,\lceil\sqrt{N}/2\rceil,\lceil\sqrt{N}\rceil,\lceil 2\sqrt{N}\rceil,\lceil 4\sqrt{N}\rceil\}$,其中 N 为样本数量,而 p 值表示选取样本点的比例,其范围为(0.1~0.9),将通过 Grid search 方法来确定最优组合.

图 2 展示了不同项目的 K 和 p 组合,相应的 F-measure 分布情况. 由图 2 可知,我们能够根据最优的 F-measure 找到合适的参数组合.

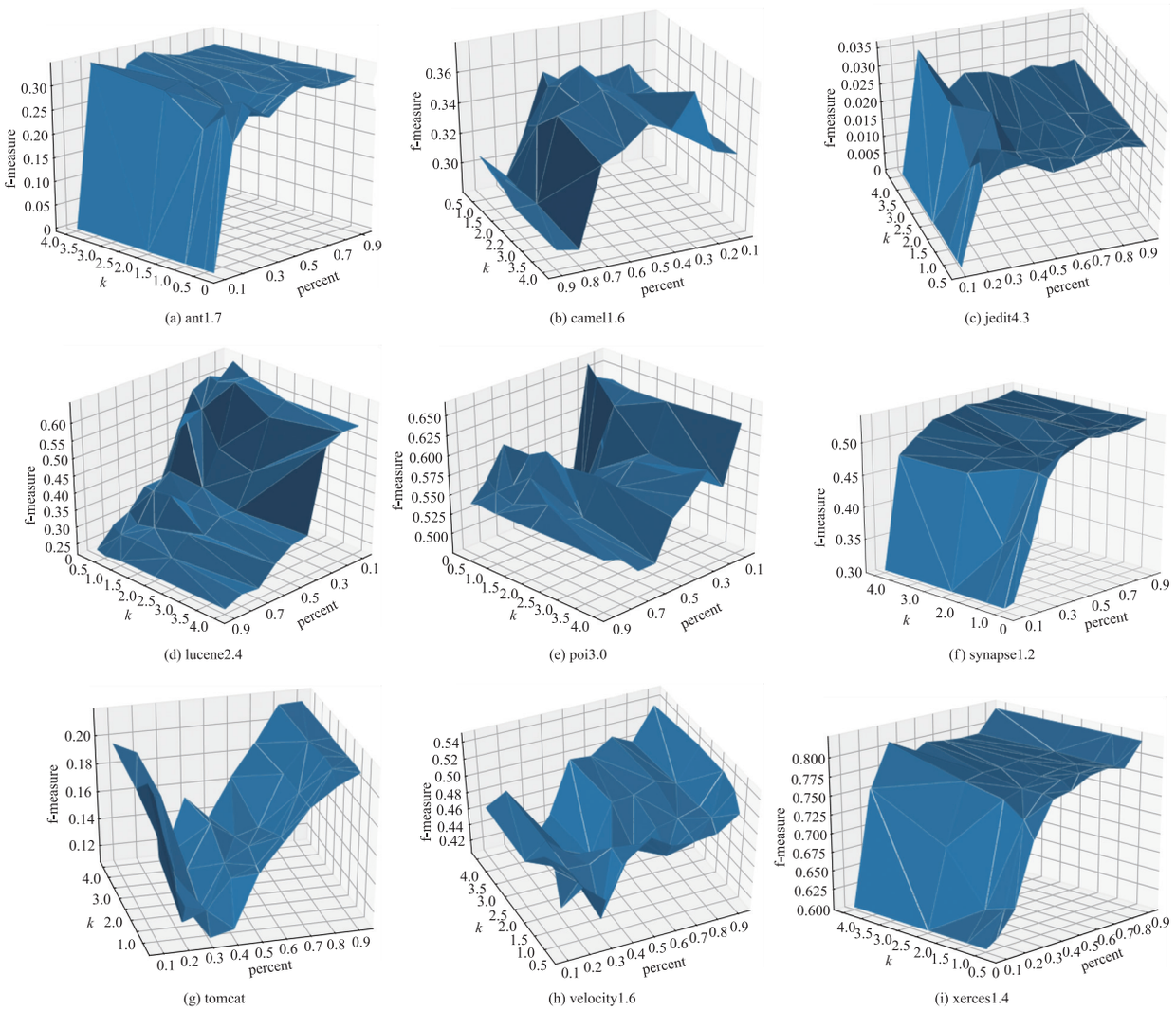

图 2　不同 K 和 p 组合的 F-measure

Fig. 2　F-measure under different K and p

如图 3 所示,文章进一步讨论了 K 和 p 在相同数据集上的不同分类器下 F-measure 的值变化情况. 以 xerces 为例,图中结果表明,当 F-measure 值达到最高时,不同分类器在当前数据集上的 K 和 p 的组合基本保持一致,这说明经过本文方法所选择的训练数据已经达到最优,且能够适应 CPDP 中各类常用的分类器.

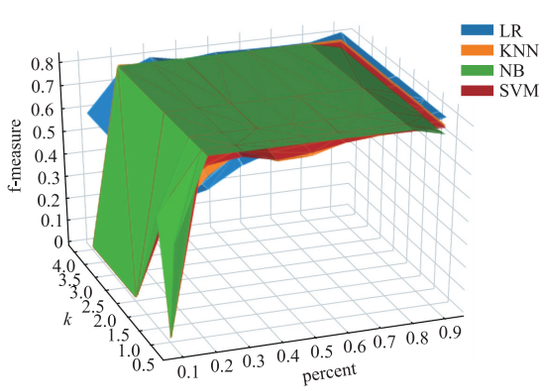

图 3　xerces 不同分类器的 F-measure

Fig. 3　F-measure of xerces under different classifiers

5　结论

本文围绕一对一的同构跨项目缺陷预测展开研究,主要针对源项目训练数据选择的问题,提出基于 JS 散度和相对密度的跨项目缺陷预测方法. 该方法首先利用 JS(Jensen-Shannon divergence)散度选择与目标项目最相似的源项目;其次,提出基于相对密度的源项目数据选择方法;最后,采用 CPDP 中常见的分类器构建预测模型,并用于目标项目进行验证. 实验结果表明,在最大程度利用源项目的情况下,本方法不仅能够提高缺陷预测模型的性能,同时对不同分类器表现出较高的适应性.

后续工作中,我们将进一步在更多的软件缺陷数据集上验证方法的有效性,并对方法进行扩展,使其应用于多对一跨项目缺陷预测.

[参考文献]

[1] JURECZKO M,MADEYSKI L. Cross-project defect prediction with respect to code ownership model:an empirical study[J]. E-informatica software engineering journal,2015,9(1):21-35.

[2] TURHAN B. On the dataset shift problem in software engineering prediction models[J]. Empirical software engineering,2012,17(1/2):62-74.

[3] ZIMMERMANN T,NAGAPPAN N,GALL H,et a1. Cross-project defect prediction:a large scale experiment on data vs. Domain vs. process[C]//In Proceedings of the 7th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT International Symposium on Foundations of Software Engineering. Amsterdam:Netherland,2009:91-100.

[4] TURHAN B,MENZIES T,BENER A B,et a1. On the relative value of cross-company and within-company data for defect prediction[J]. Empirical software engineering,2009,14(5):540-578.

[5] HE P,LI BING,ZHANG D G,et a1. Simplification of training data for cross-project defect prediction[J]. arXiv:1405.0773,2014.

[6] PETERS F,MENZIES T,MARCUS A. Better cross company defect prediction[C]//Mining Software Repositories. San Francisco:IEEE,2013:409-418.

[7] 李勇,黄志球,王勇,等. 基于多源数据的跨项目软件缺陷预测方法[J]. 吉林大学学报(工学版),2016,46(6):2034-2041.

[8] NAM J,PAN S J,KIM S. Transfer defect learning[C]//Proceedings of the International Conference on Software Engineering. San Francisco:IEEE,2013:382-391.

[9] ZHAO H Q,ZENG X P,ZHANG J S. Adaptive reduced feedback FLNN filter for active control of nonlinear noise processes[J]. Signal processing,2010,90(3):834-847.

[10] MA Y,LUO G,ZENG X,et al. Transfer learning for cross-company software defect prediction[J]. Information & software technology,2012,54(3):248-256.

[11] PENG L,YANG B,CHEN Y,et al. Data gravitation based classification[J]. Information ences,2009,179(6):809-819.

[12] CHEN L,FANG B,SHANG Z,et al. Negative samples reduction in cross-company software defects prediction[J]. Information & software technology,2015,62:67-77.

[13] DAI W Y,YANG Q,XUE G,et al. Boosting for transfer learning[C]//Proceedings of the 24th International Conference on Machine Learning-ICML'07. Oregon:USA,2007.

[14] 周志华. 机器学习[M]. 北京:清华大学出版社,2016.

[15] LIN J. Divergence measures based on the Shannon entropy[J]. IEEE transactions on information theory,2002,37(1):145-151.

[16] HERSHEY J R,OLSEN P A. Approximating the Kullback Leibler divergence between gaussian mixture models[C]//IEEE International Conference on Acoustics. Honolulu HI:IEEE,2007.

[17] WANG Q,KULKARNI S R,VERD S. Divergence estimation for multidimensional densities via-nearest-neighbor distances[J]. IEEE transactions on information theory,2009,55(5):2392-2405.

[18] O'HAGAN,ADRIAN A,MURPHY T B,GORMLEY I C,et al. Clustering with the multivariate normal inverse Gaussian distribution[J]. Computational stats & data analysis,2016,93(C):18-30.

[19] JURECZKO M,MADEYSKI L. Towards identifying software project clusters with regard to defect prediction[C]//Proceedings of the 6th International Conference on Predictive Models in Software Engineering-PROMISE' 10. Timisoars:Romania,2010.

[20] GRETTON A,SCHÖLKOPF B,HUANG J. Correcting sample selection Bias by unlabeled date[J]. Advances in neural information processing systems,2007,19:601-608.

[21] QIU S,LU L,JIANG S. Multiple components weights model for cross-project defect prediction[J]. IET software,2018,12(4):345-355.

[责任编辑:陆炳新]