增加度量元的迁移学习跨项目软件缺陷预测

娄丰鹏[1],吴 迪[2],荆晓远[3],吴 飞[3]
(1.南京邮电大学 计算机学院,江苏 南京 210003;
2.武汉大学 计算机学院 软件工程国家重点实验室,湖北 武汉 430072;
3.南京邮电大学 自动化学院,江苏 南京 210003)

摘 要:目前,结合机器学习方法和软件缺陷预测技术自动地学习模型来发现软件中的缺陷,已经成为跨项目缺陷预测的主要方法。由于源项目和目标项目之间的特征分布差异,跨项目相关性预测的表现通常较差。针对该问题,可以使用从源项目中提取知识并将其转移到目标项目的转移学习技术来提高预测性能,并提出了一种增加度量元的迁移学习方法进行跨项目的软件缺陷预测。该方法首先使用分类器对数据集进行一次项目内预测,并将预测结果作为新的度量元加入数据集。然后采用迁移学习方法将源项目中提取的知识转移至目标项目,并使用分类器预测目标项目。在AEEEM数据集上的实验结果表明,该算法提高了跨项目软件缺陷预测效率。

关键词:跨项目;机器学习;软件缺陷预测;迁移学习;分类器
中图分类号:TP301.6 **文献标识码**:A **文章编号**:1673-629X(2018)07-0103-05
doi:10.3969/j.issn.1673-629X.2018.07.022

Cross-project Software Defect Prediction Based on Transfer Learning with Metrics

LOU Feng-peng[1],WU Di[2],JING Xiao-yuan[3],WU Fei[3]
(1.School of Computer,Nanjing University of Posts and Telecommunications,Nanjing 210003,China;
2.State Key Laboratory of Software Engineering,School of Computer,Wuhan University,Wuhan 430072,China;
3.School of Automation,Nanjing University of Posts and Telecommunications,Nanjing 210003,China)

Abstract: The combination of machine learning method and software defect prediction technology to automatically learn the model to find the software defects has become the main method of cross-project defect prediction in recent years.However,the performance of cross-project prediction is generally poor largely due to feature distribution differences between the source and target projects.In this cases,we propose a method to add the prediction results as a metric to the original data set to form a new data set for cross-project software defect prediction.First,a classifier is used to predict the data set in a project,and the predicted result is added as a new metric to the data set.Second,a transfer learning method is applied to transfer knowledge from original source project to target project,and a classifier is used to predict target project,thus effectively improving the prediction accuracy.The experiment on AEEEM data set shows that the proposed method significantly improves cross-project prediction performance.

Key words: cross-project; machine learning; software defect prediction; transfer learning; classifier

0 引 言

软件缺陷通常导致系统发生故障,进而造成财政和金融损失。软件在正式发布之前,可以通过不同级别的测试来检验和删除这些缺陷。因此,软件缺陷预测是保证系统正常运行的一个至关重要的步骤。

在软件工程领域,软件缺陷预测已经成为一个重大的研究课题,引起了很多学者的高度重视[1-2]。近年来,已经提出了许多有效的软件缺陷预测方法[3-5]。这些方法是通过从软件仓库中挖掘数据集,然后使用机器学习分类器预测一个学习模型[6],通过该模型可以对软件缺陷进行预测[7]。近年来比较流行的机器学习分类器有很多,如决策树(decision tree,DT)、随机森

收稿日期:2017-08-12 **修回日期**:2017-12-28 **网络出版时间**:2018-03-07
基金项目:国家自然科学基金(61272273)
作者简介:娄丰鹏(1991-),女,研究生,研究方向为信息安全、机器学习与数据挖掘;荆晓远,教授,博导,研究方向为模式识别、图像与信号处理、信息安全、机器学习与数据挖掘。
网络出版地址:http://kns.cnki.net/kcms/detail/61.1450.TP.20180307.1427.052.html

林(random forest, RF) 、逻辑回归(logistic regression, LR) 、支持向量机模型(support vector machine, SVM) [8-9] 等。但是,大多数方法是在项目内部进行评估的,新项目通常没有足够的缺陷数据来构建预测模型。对于一个新项目或者是具有有限训练数据的新项目,最好通过使用现有源项目中足够的训练数据来学习预测模型,然后将模型应用于某些目标项目,称之为跨项目。因此,跨项目缺陷预测是必要的。但是,由于源项目和目标项目之间的特征分布差异,导致跨项目软件缺陷预测的性能通常较差。

针对上述问题,文中采用迁移学习方法(transfer component analysis, TCA) [7] 查找源项目与目标项目的相似性,并提出一种通过增加度量元来提高基于迁移学习的跨项目缺陷预测性能的方法(MTCA) 。该方法首先使用分类器对数据集进行一次项目内快速预测,将预测结果作为新的度量元加入数据集构成新的数据集。然后,采用迁移学习将源项目中提取的知识转移至目标项目,并使用分类器预测目标项目。最后在 AEEEM 数据库[10] 上进行验证。

1 机器学习算法以及缺陷预测过程

1.1 机器学习分类器算法研究

逻辑回归算法广泛应用于数据挖掘、数据分类,同时也在支持概率类型结果输出的方面得到大量应用。逻辑回归算法是在线性回归的基础上,加入逻辑函数,使用输入变量线性加权实现分类,最终输出概率估计。

支持向量机算法[8-9] 是基于核的算法,主要是把输入数据映射到一个高维空间,使输入数据由线性不可分以最大化分类间隔构建最优分割超平面,从而提高学习机的泛化能力。对于分类,支持向量机算法根据空间中的样本计算该空间的决策曲面,由此确定该空间中未知样本的所属类别。

随机森林算法是多个决策树分类器的集成,其输出的类别是由个别决策树输出的类别的众数决定的。随机森林训练和预测速度快,对训练数据的容错能力可以进行有效估计,当数据集中的数据缺失很多时依旧可以保持精度不变,能够在分类过程中生成一个泛化误差较小的内部无偏估计。

近邻算法中,所选择的邻居都是已经得到正确分类的数据。该方法在定类决策上只依据最邻近的一个或者几个样本的类别来决定待分样本所属的类别,最近邻样本中的大多数属于哪个类别,该样本就属于哪个类别。

1.2 缺陷预测过程

图 1 是一般的软件缺陷预测过程。

在项目内预测中,训练样本集和测试样本集来自同一个项目。对于跨项目预测,训练样本集来自一个项目(源) ,测试样本集来自另一个项目(目标) 。

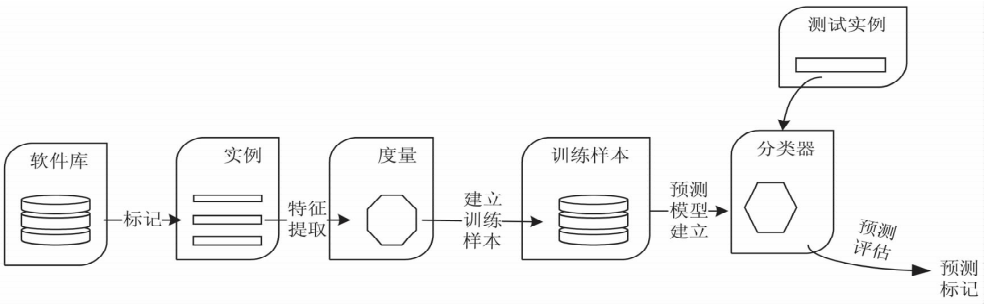

图 1 软件缺陷预测过程

如图 1 所示,该过程首先是收集软件并标记每个软件发布后的缺陷数量,如果一个软件存在至少一个缺陷,则表明该软件存在缺陷,否则为无缺陷。通过提取复杂度等度量用于机器模型训练分类器,然后通过分类器预测一个新的样本的缺陷情况。

2 迁移学习与文中算法

2.1 迁移学习以及模型定义

近年来,机器学习和数据挖掘技术引起了越来越多的关注[11] 。在训练样本和测试样本具有相同的特征空间和分布时,大多数机器学习方法可以获得良好的性能[11] 。当特征空间和分布发生变化时,学习模型需要重建。在这种情况下,有必要重新收集训练数据并再次标记。通常,重建学习模型是昂贵的,并且标注新的训练数据需要相当大的努力。迁移学习通过转移从相关但不同领域提取的知识来解决这些问题,这可以被认为是相关性预测中的源项目,而在目标领域建立精确的预测模型,可以被视为目标项目[12] 。

迁移学习的模型定义如下:

设 X_T 为目标样本空间, X_S 为源样本空间,目标样本空间也即目标空间,就是想要去分类的样本空间。设 $Y = \{0,1\}$ 为类空间,训练数据也就是: $T \subseteq \{(X = X_S) \times Y\}$ 。

测试数据: $S = \{(x_i^t)\}$, $x_i^t \in X_T, i = 1,2,\cdots,k$ 。其中测试数据是未标记的,将训练数据表示为: $T_a = \{(x_i^a, c(x_i^a))\}$, $x_i^a \in X_S, i = 1,2,\cdots,n$ 。其中 $c(x)$ 代

表样本数据 x 真实属于的类别，T_a 和测试数据 S 是属于不同分布的。现在的任务就是给定源数据 T_a，训练出一个分类器，使得在测试数据 S 上的分类误差最小。

2.2　TCA(transfer component analysis)

TCA 是一种基于特征的迁移学习方法，使用 MMD(maximum mean discrepancy) [13-14] 将处于不同数据分布的源空间和目标空间一起映射到一个高维的再生核希尔伯特空间。在该空间中，最小化源数据和目标数据距离，同时最大程度地保留源数据和目标数据各自的内部属性，那么跨空间的差异可能会大大降低。因此，通过这个高维空间中的新表示，可以应用标准的机器学习方法来训练源空间中的分类器或回归模型，以在目标空间中使用。TCA 算法描述如下:

输入: 源空间训练样本集 T_a，目标空间测试样本集 T_b，训练数据 M，测试数据 N。

预处理: 对样本集进行预处理(如归一化、降维等) 。

(1) 构造 $\boldsymbol{L}$ 矩阵。

$\boldsymbol{L} = -\frac{1}{M * N} * \boldsymbol{I}$，$\boldsymbol{I}$ 为 $M + N$ 阶全 1 矩阵。

$\boldsymbol{L}(1:M,1:M) = \frac{1}{M^2} * \boldsymbol{I}_1$，$\boldsymbol{I}_1$ 为 M 阶全 1 矩阵。

$\boldsymbol{L}(M+1:M+N,M+1:M+N) = \frac{1}{N^2} * \boldsymbol{I}_2$，$\boldsymbol{I}_2$ 为 N 阶全 1 矩阵。

(2) 构造 $\boldsymbol{H}$ 矩阵。

$\boldsymbol{H} = E_1 - \frac{1}{M+N} * \boldsymbol{I}_3 * \boldsymbol{I}_4$，$\boldsymbol{E}_1$ 为 $M+N$ 阶单位阵，$\boldsymbol{I}_3$ 为 $M+N$ 行 1 列的全 1 矩阵，$\boldsymbol{I}_4$ 为 1 行 $M+N$ 列全 1 矩阵。

(3) 构造核函数矩阵 $\boldsymbol{K}$。

根据 T_a 和 T_b，使用常用核函数计算 $\boldsymbol{K}$。

(4) 构造矩阵 $\boldsymbol{W}$。

求 $(\boldsymbol{KLK} + \mu\boldsymbol{I})^{-1}\boldsymbol{KHK}$ 的前 m 个特征值，即为 $\boldsymbol{W}$ 矩阵。

(5) 映射样本数据: $\boldsymbol{W}' * \boldsymbol{T}$。

输出: 源训练样本和目标测试样本的降维数据。

2.3　文中算法流程

在上一节的基础上，得到降维之后的训练样本数据运用传统机器学习方法训练源空间的分类器，并对目标空间的测试数据进行缺陷预测。结合前面介绍的方法，得到文中算法流程，如图 2 所示。

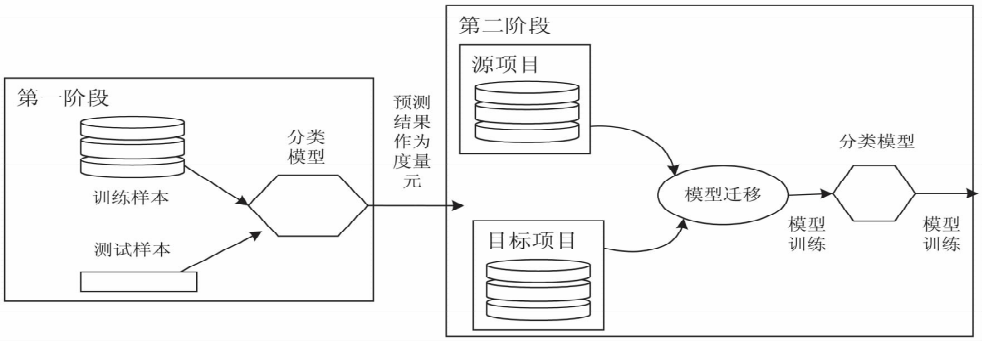

图 2　文中算法流程

图中第一阶段代表第一次工程内预测，将预测结果作为工程的度量元加入数据集，通过数据集预处理得到新数据集作为第二阶段的输入数据[15]; 第二阶段运用 TCA 算法，结合复杂的分类模型进行跨工程预测，得到最终的预测结果。算法 MTCA 第二阶段伪代码描述如下:

输入: 源空间训练样本集 X_S，目标空间测试样本集 X_T。

预处理: 对样本集进行预处理(如归一化、降维等) 。

步骤 1: 将 X_S 和 X_T 利用 TCA 算法进行处理，得到迁移后的新的数据集 X_S' 和 X_T'。

步骤 2: 使用 SVM 分类器对每一个测试样本实例进行预测分类。

输出: 目标测试样本的预测标记。

3　实　验

3.1　实验数据库

实验是在 AEEEM [10] 数据库上进行的，该数据库包含 5 个工程，表 1 介绍了各个工程的静态代码度量和缺陷数等特征[16]。

表 1　AEEEM 数据集

数据库	缺陷样本数	样本总数	特征数	缺陷样本占比/%
EQ	129	324	61	39.81
JDT	206	997	61	20.66
LC	64	691	61	9.26
ML	245	1 862	61	13.16
PDE	209	1 497	61	13.96

3.2 评估度量

通过 Precision(精确度) 、Recall(召回率) 以及 F - measure(综合评估) 三个指标来全面评估各个方法的性能,这三个指标是评估分类模型最具代表性的度量[17],一个好的预测模型希望实现较高的 Recall 和 Precision 值。

根据表 2,Precision 和 Recall 可以定义为:

$$\text{Precision} = \frac{A}{A + C} \tag{1}$$

$$\text{Recall} = \frac{A}{A + B} \tag{2}$$

表 2 四种预测结果

	预测为相关	预测为不相关
真实为相关	A	B
真实为不相关	C	D

但是,Recall 和 Precision 两个指标之间存在着权衡。因此,需要对召回率和精确度进行综合评估,则 F - measure 定义为:

$$F - \text{measure} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

这四种评价指标值都在 0~1 之间,一个好的缺陷预测模型应该会有较高的 Precision、F -measure 和 Recall 值。而且 F -measure 是综合性评价指标,更加重要。

3.3 实验结果与分析

文中使用一对一的跨工程预测(即仅使用一个源项目预测一个目标项目) ,使用 AEEEM 库构建一个跨项目组合,例如: EQ → JDT、EQ → LC、LC → PDE 等。实验步骤为: 首先是 AEEEM 库中的 5 个工程各自做一次工程内缺陷预测,分类器分别是 LR、RF、NN,将预测结果作为工程的度量元加入数据集,每个原始数据集加入一个特征,即构成了三组新的数据集,即(EQ_LR、JDT_LR、ML_LR、LC_LR、JDT_LR) 为一组新数据集,(EQ_RF、JDT_RF、ML_RF、LC_RF、JDT_RF) 为一组新数据集,(EQ_NN、JDT_NN、ML_NN、LC_NN、JDT_NN) 为一组新数据集。每组新数据集进行跨项目组合并迁移,然后进行跨工程的软件缺陷再预测,二次预测使用一个鲁棒性更强的分类器,文中使用 SVM,这一算法流程简称 MTCA。在对比算法 TCA 时,为了保证实验的可靠性,使用 SVM 分类器。对 TCA 方法与 MTCA 方法做对比,TCA 使用的是原始数据集,MTCA 使用的是 EQ、JDT、LC、ML、PDE 三组新数据集,随机进行 20 次求平均,结果如表 3 所示。

表 3 在 AEEEM 数据库上的实验结果

源 → 目标	Precision				Recall				F -measure			
	RF	NN	LR	TCA	RF	NN	LR	TCA	RF	NN	LR	TCA
EQ → JDT	0.51	0.48	0.48	0.42	0.63	0.63	0.54	0.61	0.60	0.59	0.53	0.56
LC → JDT	0.33	0.25	0.42	0.27	0.51	0.64	0.32	0.50	0.46	0.48	0.44	0.43
ML → JDT	0.23	0.40	0.36	0.25	0.56	0.48	0.48	0.30	0.44	0.41	0.33	0.31
PDE → JDT	0.38	0.23	0.44	0.19	0.60	0.48	0.43	0.39	0.42	0.42	0.43	0.39
JDT → EQ	0.50	0.49	0.49	0.47	0.50	0.51	0.33	0.53	0.52	0.51	0.35	0.50
LC → EQ	0.48	0.46	0.42	0.81	0.35	0.42	0.53	0.27	0.37	0.43	0.51	0.31
ML → EQ	0.51	0.32	0.33	0.33	0.57	0.53	0.49	0.47	0.55	0.47	0.44	0.45
PDE → EQ	0.38	0.31	0.45	0.30	0.42	0.48	0.55	0.51	0.48	0.43	0.53	0.39
EQ → LC	0.22	0.38	0.37	0.13	0.67	0.44	0.45	0.66	0.47	0.42	0.43	0.36
JDT → LC	0.53	0.55	0.46	0.42	0.54	0.34	0.52	0.50	0.48	0.37	0.30	0.34
ML → LC	0.46	0.22	0.35	0.18	0.62	0.48	0.62	0.52	0.58	0.27	0.41	0.29
PDE → LC	0.40	0.21	0.34	0.23	0.51	0.27	0.56	0.42	0.48	0.25	0.50	0.36
EQ → ML	0.47	0.34	0.31	0.23	0.59	0.30	0.73	0.53	0.42	0.31	0.45	0.42
JDT → ML	0.24	0.25	0.49	0.17	0.52	0.59	0.45	0.51	0.42	0.46	0.34	0.36
LC → ML	0.42	0.33	0.43	0.21	0.33	0.36	0.36	0.24	0.35	0.25	0.37	0.23
PDE → ML	0.56	0.48	0.54	0.12	0.65	0.55	0.63	0.60	0.63	0.53	0.61	0.34
EQ → PDE	0.27	0.39	0.25	0.26	0.61	0.32	0.57	0.56	0.48	0.33	0.46	0.45
JDT → PDE	0.28	0.29	0.25	0.17	0.57	0.60	0.55	0.43	0.47	0.50	0.44	0.33
LC → PDE	0.39	0.24	0.15	0.15	0.35	0.21	0.46	0.31	0.36	0.39	0.33	0.36

ML → PDE	0.55	0.50	0.53	0.36	0.46	0.50	0.48	0.37	0.48	0.50	0.49	0.29

分析表 3 可知,MTCA 在各个数据库上的缺陷预测效果普遍好于 TCA,尤其是 F -measure 评价指标,相较其他方法优势明显。F -measure 作为综合评价指标,从该值的观测中就能看出算法的总体性能,而 MTCA 在 F -measure 上较 TCA 要高出很多,也说明了该方法的优势。但是,也存在一些结果相比较低的,原因是加入新的度量元时可能引入了过多的错误数据,导致缺陷预测模型向着错误的方向构建。总体上结果还是普遍较好。

4　结束语

针对当前软件缺陷预测模型中机器学习算法对预测模型性能的影响问题,提出了一种通过增加度量元来提高基于迁移学习的跨项目缺陷预测性能的方法。首先使用分类器对数据集进行一次项目内预测,并将预测结果作为新的度量元加入数据集; 然后采用迁移学习方法将源项目中提取的知识转移至目标项目,并使用分类器预测目标项目,从而提高了跨项目软件缺陷预测效率。通过对比验证了该方法的有效性。如何合理选择分类器以进一步提高缺陷预测模型的性能指标是下一步研究的问题。

参考文献:

[1] RAHMAN F, POSNETT D, HERRAIZ I, et al. Sample size vs. bias in defect prediction [C]//Joint meeting on foundations of software engineering. [s.l.]: ACM, 2013: 147-157.

[2] RAHMAN F, KHATRI S, BARR E T, et al. Comparing static bug finders and statistical prediction [C]//Proceedings of the 36th international conference on software engineering. Hyderabad, India: ACM, 2014: 424-434.

[3] HASSAN A E. Predicting faults using the complexity of code changes [C]//31st international conference on software engineering. Vancouver, BC, Canada: IEEE, 2009: 78-88.

[4] KIM S, JR E J W, ZHANG Yi. Classifying software changes: clean or buggy? [J]. IEEE Transactions on Software Engineering, 2008, 34(2): 181-196.

[5] MENZIES T, GREENWALD J, FRANK A. Data mining static code attributes to learn defect predictors [J]. IEEE Transactions on Software Engineering, 2006, 33(1): 2-13.

[6] 戴文渊. 基于实例和特征的迁移学习算法研究 [D]. 上海: 上海交通大学, 2009.

[7] NAM J, PAN S J, KIM S. Transfer defect learning [C]//35th international conference on software engineering. San Francisco, CA, USA: IEEE, 2013: 382-391.

[8] ELISH K O, ELISH M O. Predicting defect-prone software modules using support vector machines [J]. Journal of Systems & Software, 2008, 81(5): 649-660.

[9] TANTITHAMTHAVORN C, MCINTOSH S, HASSAN A E, et al. Comments on "researcher bias: the use of machine learning in software defect prediction" [J]. IEEE Transactions on Software Engineering, 2016, 42(11): 1092-1094.

[10] FENTON N E, NEIL M. Software metrics: roadmap [C]// Conference on the future of software engineering. [s.l.]: IEEE, 2000: 357-370.

[11] PAN S J, YANG Qiang. A survey on transfer learning [J]. IEEE Transactions on Knowledge & Data Engineering, 2010, 22(10): 1345-1359.

[12] 刘英博, 王建民. 面向缺陷分析的软件库挖掘方法综述 [J]. 计算机科学, 2007, 34(9): 1-4.

[13] BORGWARDT K M, GRETTON A, RASCH M J, et al. Integrating structured biological data by kernel maximum mean discrepancy [J]. Bioinformatics, 2006, 22(14): e49-e57.

[14] 皋　军, 黄丽莉. 最大局部加权均值差异嵌入 [J]. 电子学报, 2013, 41(8): 1462-1468.

[15] 陈家强. 软件缺陷预测中数据预处理技术研究 [D]. 南京: 南京大学, 2014.

[16] 陈　翔, 顾　庆, 刘望舒, 等. 静态软件缺陷预测方法研究 [J]. 软件学报, 2016, 27(1): 1-25.

[17] RASTKAR S, MURPHY G C, MURRAY G. Summarizing software artifacts: a case study of bug reports [C]//ACM/IEEE international conference on software engineering. [s.l.]: ACM, 2010: 505-514.

(上接第 102 页)

l.]: [s.n.], 2015: 918-927.

[10] DWORK C, MCSHERRY F, NISSIM K, et al. Calibrating noise to sensitivity in private data analysis [C]//3rd conference on theory of cryptography. [s.l.]: [s.n.], 2006: 265-284.

[11] DWORK C. Differential privacy in new settings [C]//Proceedings of the twenty-first annual ACM-SIAM symposium on discrete algorithms. Austin, Texas, USA: [s.n.], 2010: 174-183.

[12] SWEENEY L. Achieving k-anonymity privacy protection using generalization and suppression [J]. International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, 2002, 10(5): 571-588.

[13] 张啸剑, 王　淼, 孟小峰. 差分隐私保护下一种精确挖掘 top-k 频繁模式方法 [J]. 计算机研究与发展, 2014, 51(1): 104-114.

[14] 欧阳佳, 印　鉴, 刘少鹏, 等. 一种有效的差分隐私事务数据发布策略 [J]. 计算机研究与发展, 2014, 51(10): 2195-2205.

[15] 薛寿豪, 张正道. 基于箱聚类的差分隐私直方图发布方法研究 [J]. 计算机应用研究, 2014, 31(12): 3700-3703.