文章编号：2096-1618(2018)03-0286-04

基于 NPE-SVM 的软件缺陷预测模型

王玉红[1]， 范 菁[1]， 雷 敏[2,3]， 孙汇中[2]

(1. 云南民族大学电气信息工程学院，云南 昆明 650000；2. 北京邮电大学信息安全中心，北京 100876；3. 贵州大学贵州省公共大数据重点实验室，贵州 贵阳 550025)

摘要：针对软件缺陷预测中数据集的类不均衡、高维、小采样以及非线性降维问题，提出基于领域保持嵌入支持向量机的软件缺陷预测模型。模型采用 NPE 算法对数据集进行降维处理，通过将 NPE 算法中奇异的广义特征计算转化为两个特征分解问题，得到了更准确的稳健解，有效规避了属性约减后导致的预测精度下降问题。选用支持向量机作为基础分类器，仿真实验结果表明，与其他方法相比，预测模型的查全率及 F-measure 值指标显著提高了2% ~4%。

关 键 词：软件缺陷；领域保持嵌入；机器学习；模式识别；流行学习

中图分类号：TP311.53 **文献标志码**：A

doi：10.16836/j.cnki.jcuit.2018.03.011

0 引言

软件缺陷预测，即识别软件系统中含缺陷的高风险模块，合理分配软件测试资源[1]。软件缺陷预测已成为软件系统安全中的一项重要的技术，然而，实际应用中的软件缺陷测试集都是高维的，难以识别。且研究学者 Boehm 指出软件缺陷分布符合 2-8 原则，即80%的缺陷包含在20%的软件模块中，表明软件缺陷数据集也存在严重的类不均衡问题。因此，降低软件缺陷测试集维度、获取软件缺陷特征在软件缺陷预测中显得尤为重要。

目前，软件缺陷预测领域的研究人员采用最多的就是机器学习的方法，如聚类分析、人工神经网络、线性判别分析等，但都存在一些问题：聚类分析可重复性差且成本较高；人工神经网络所需样本较多；线性判别分析预测精度比较低[2-4]。针对上述问题，采用领域保持嵌入(neighborhood preserving embedding，NPE)算法，通过转化 NPE 算法中奇异的广义特征计算问题改进 NPE 算法，发现低维空间中数据的真实结构。由于支持向量机(support vector machine，SVM)在处理小样本、非线性及高维模式识别问题中具有独特的优势，因此选用 SVM 进行分类预测。解决软件缺陷数据集的类不均衡及高维、小采样问题。

1 NPE 算法

1.1 NPE 算法

NPE 算法认为样本空间中任意样本点都可由其领域内的 K 个样本点线性表示，当数据集嵌入到更低维度时，将非线性数据特征转化为多个线性特征，保留了数据的全局几何结构信息[5-6]。

第一步：寻找邻近点。

基于欧式距离选取每个样本点领域内的 K 个样本点为邻近点。由于在模式识别问题中数据集标签已知，借助标签信息选取标签相同的数据点为该样本的 K 个邻近点。

第二步：计算最优重构权值矩阵。

根据欧氏距离最短原则重构向量，重构误差最小为约束，求得最优权值矩阵：

$$\boldsymbol{d}_i = \| x_i - x_j \|$$

$$\arg\min_{\boldsymbol{W}} \varepsilon(w) = \arg\min_{\boldsymbol{W}} \sum_i \| x_i - \sum_j w_{ij} x_j \|^2$$

上述的最小化有两个条件的限制：权值矩阵应满足稀疏性条件 $W_{ij} = 0$(x_j 不是 x_i 的邻近点之一)；归一化条件 $\sum_{j=1}^{k} w_{ij} = 1$。

这样 W 可以求得它的闭解形式：

$$W_{ij} = \frac{\sum_{t=1}^{k} \boldsymbol{G}_{jt}^{-1}}{\sum_{p=1}^{k}\sum_{q=1}^{k} G_{pq}^{-1}}$$

线性降维本质上是寻找一组线性投影方向 $\boldsymbol{g}_0, \boldsymbol{g}_1, \cdots, \boldsymbol{g}_{d-1}$，对应的线性投影矩阵为 $\boldsymbol{G} = [\boldsymbol{g}_0, \boldsymbol{g}_1, \cdots, \boldsymbol{g}_{d-1}] \in R^{D\times d}$，使得低维嵌入表示 $y_i = \boldsymbol{G}^{\mathrm{T}} \times x_i$ 是原始数据 x_i 期望的低维表示。

第三步：获取低维嵌入表示。

保持重构权重不变，将样本点向低维度空间映射时，保留了局部几何结构。y_i 为样本点 x_i 在低维空间中的映射，y_j 为样本点 x_j 在低维空间中的映射。NPE

收稿日期：2018-03-20

基金项目：国家自然科学基金资助项目(61540063)；云南省教育厅资助项目(2017ZDX045)；贵州省公共大数据重点实验室开放课题基金资助项目(2017BDKFJJ017)

算法以原始高维空间的重构权重为约束,获取低维空间样本表示,优化目标函数如下:

$$\arg\min \varphi(Y)=\sum_{i=1}^{N}\| y_i-\sum_{j=1}^{k}w_{ij}y_j \|^2$$

$$\text{S.t}\quad YY^{\mathrm{T}}=c$$

其中,c 是一个常量,可使得 $\boldsymbol{g}$ 为一单位向量。又 $y_i=\boldsymbol{G}^{\mathrm{T}}x_i$,经过简单的代数推导,可将上述条件化为

$$\arg\min_{\boldsymbol{g}} \boldsymbol{g}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{g}$$

$$\text{S.t.}\quad \boldsymbol{g}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{g}=c$$

其中 $\boldsymbol{M}=(\boldsymbol{I}-\boldsymbol{W})(\boldsymbol{I}-\boldsymbol{W})^{\mathrm{T}}$,$\boldsymbol{I}$ 是单位矩阵。利用拉格朗日乘子法,将求解线性投影向量 $\boldsymbol{G}$ 问题转化为求解广义特征值问题。

$$\boldsymbol{X}\boldsymbol{M}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{g}=\lambda\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{g}$$

设 $\boldsymbol{g}_0,\boldsymbol{g}_1,\cdots,\boldsymbol{g}_{d-1}$ 为所得的特征向量,按照对应特征值大小从小到大排列,即 $\lambda_0<\lambda_1<\cdots<\lambda_{d-1}$ 排序,最后可得到低维空间的映射坐标。

1.2　NPE 算法改进

因数据集特征矩阵 $\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}$ 的秩 $\mathrm{rank}(\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}})=\mathrm{rank}(\boldsymbol{X})\leqslant\min(D,N)=N$。而 $\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}$ 是 D 维方阵,所以当样本点数目较少但维度较高的条件下,特征矩阵就变成了奇异矩阵,使得 NPE 算法不可行。

改进 NPE 算法中奇异矩阵的特征计算问题,将特征向量的求解转化为计算两矩阵的特征分解,在简化广义特征分解计算问题的同时,保留了有用的原始判别条件,求得了更加稳定的特征向量。

结合 NPE 算法最后一步中线性投影方向在高维小采样下求解,具体过程如下:

(1)计算矩阵

$\{\boldsymbol{X}[(1-\boldsymbol{W})^{\mathrm{T}}\boldsymbol{I}]\}^{\mathrm{T}}\boldsymbol{X}[(1-\boldsymbol{W})^{\mathrm{T}}\boldsymbol{I}]\in\boldsymbol{R}^{2N\times 2N}$ 的特征分解:

$$\{\boldsymbol{X}[(1-\boldsymbol{W})^{\mathrm{T}}\boldsymbol{I}]\}^{\mathrm{T}}\boldsymbol{X}[(1-\boldsymbol{W})^{\mathrm{T}}\boldsymbol{I}]=[\boldsymbol{V}_r\tilde{\boldsymbol{V}}_r]\begin{bmatrix}[\boldsymbol{I}]_1 & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}\boldsymbol{V}_r^{\mathrm{T}}\\ \tilde{\boldsymbol{V}}_r^{\mathrm{T}}\end{bmatrix}$$

由该矩阵的非零特征值组成对角矩阵 $[\boldsymbol{I}]_1$,其对角阵元素由大到小依次排列。该矩阵零特征值对应的特征向量组成矩阵 $\tilde{\boldsymbol{V}}_r$,其非零特征值对应特征向量组成矩阵 $\boldsymbol{V}_r$。

(2)计算矩阵 $\boldsymbol{S}_t=\boldsymbol{X}\boldsymbol{M}\boldsymbol{X}^{\mathrm{T}}+\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\in R^{D\times D}$ 的特征分解中的矩阵 $\boldsymbol{U}_r$ 和 $\sum_I$:

$$S_t=[\boldsymbol{U}_r\tilde{\boldsymbol{U}}_r]\begin{bmatrix}\Sigma_I & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}\boldsymbol{U}_r^{\mathrm{T}}\\ \tilde{\boldsymbol{U}}_r^{\mathrm{T}}\end{bmatrix}=\boldsymbol{U}_r\Sigma_I\boldsymbol{U}_r^{\mathrm{T}}$$

其中 $\boldsymbol{U}_r=\boldsymbol{X}[(1-\boldsymbol{W})^{\mathrm{T}}\boldsymbol{I}]\boldsymbol{V}_r[\boldsymbol{I}]_I^{-1/2}$,$\Sigma_I=[I]_I$。

(3)计算矩阵

$\tilde{\boldsymbol{S}}_c=\sum_I^{-1/2}\boldsymbol{U}_r^{\mathrm{T}}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{U}_r\sum_I^{-1/2}\in R^{r\times r}$ 的特征分解:

$$\tilde{\boldsymbol{S}}_c=\boldsymbol{W}\Gamma\boldsymbol{W}^{\mathrm{T}}$$

(4)得到线性投影矩阵 $\boldsymbol{G}$ 为 $\boldsymbol{U}_r\sum_I^{-1/2}\boldsymbol{W}$ 的前 d 个单元列向量。

改进的 NPE 算法具有以下优良特性:克服了高维、小采样条件下奇异矩阵的广义特征分解问题,在不增加算法复杂度、不损失有用信息的情况下,可以有效地对高维数据进行特征分解,得到的特征向量具有完备性和稳健性。

2　基于 NPE-SVM 的软件缺陷预测模型

2.1　基于 NPE 算法预测模型

基于 NPE-SVM 算法的软件预测模型主要流程如图 1 所示[7]。

图 1　基于 NPE 算法的软件预测流程

首先,获取软件缺陷数据集;将数据集应用改进的 NPE 算法进行降维处理;其次将降维后的数据集作为下一步的输入集。由于径向基核函数(radial basis function,RBF)在软件缺陷预测过程中得到广泛应用,本文选用 RBF 函数,给定区间和步长,组成网格,进行网格搜索,并结合 10 折交叉验证选取最优参数,得到支持向量机最优参数模型,使用该模型测试和预测。

2.2　NPE 算法参数寻优

基于 NPE-SVM 的软件缺陷预测模型需确定两部分参数,即 NPE 算法参数和 SVM 参数的选择。NPE 算法中需考虑邻域大小 K 和嵌入维度 d,若 K 取值过大,邻域更趋近整体,样本数据的局部特性将无法体现。若 K 取值太小,嵌入到低维空间中数据的拓扑结构将难以保持。目前对于 K 值得选取没有明确的指导理论,大都凭经验或实验测试得到,本文选取经验值 $K=16$。而维度 d 要根据 NPE 算法中 $d<K$ 的准则,实验对比发现不同嵌入维度下软件缺陷预测的准确率,选取最高的 d 值。同样,SVM 部分需确定核函数参数 $\sigma(g)$ 和惩罚因子 C。将网格搜索和 10 折交叉验证相结合,与 d 值的选取类似,找到使 SVM 分类准确率最高的一组参数 C 和 g 的值,确定 SVM 分类器的最优超平面,进而得到训练好的 SVM 分类器。

2.3 NPE 算法模型评价

评价软件缺陷预测能力最广泛使用的是二维混淆矩阵,常用的评价指标有准确度、查准率、查全率和 *F*-measure 值(*F* 值)[8]。这是一种标准化的且公认的评价方法,所用混淆矩阵如表 1 所示。

表 1 混淆矩阵

实际类别	预测值	
	易出错模块	不易出错模块
有缺陷	正确的正例(TP)	错误的负例(FN)
无缺陷	错误的正例(FP)	正确的负例(TN)

准确度(accuracy):正确分类的样本点数目占总测试样本点的比率。计算公式为

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

查准率(precision):预测为有缺陷模块数目占有缺陷正确分类和无缺陷错误分类总数的比率,计算公式为

$$precision = \frac{TP}{TP + FP}$$

查全率(recall):实际和预测结果都为有缺陷模块数目占实际含缺陷模块总数的比率,计算公式为

$$recall = \frac{TP}{TP + FN}$$

F-measure 值(*F* 值):查全率与查准率的调和平均数,计算公式为

$$F\text{-measure} = \frac{2}{\frac{1}{precision} + \frac{1}{recall}}$$

3 实验与分析

3.1 数据来源

实验数据集源于美国航空航天局(National Aeronautics &Space Administration, NASA)的 MDP(metric data program, MDP)软件度量项目中的 13 个数据包,下载地址为 http://mdp.ivv.nasa.gov。数据包即软件系统,每个数据包有若干条数据,并且每条数据由多个属性描述。其中标识的代码属性主要有代码行数、McCabe 属性及 Halstead 属性。每条数据末尾都含标记位,代表对应软件模块有无缺陷。

3.2 实验验证与结果分析

文中选 2.3 各指标评价模型,采用网格搜索结合 10 折交叉验证进行实验[9]。将数据集均分成 10 组,将每个子集作一次验证集,其余子集作为训练集对 SVM 分类器训练。计算每组网格参数下分类准确率的平均值作为该组的评价指标。循环遍历整个网格区域,找到使 SVM 分类准确率最高的一组参数作为 SVM 分类器的最优参数。使用相同的实验数据,NPE-SVM 软件缺陷预测算法在不同嵌入维度下的预测结果如图 2 所示。由图可知,当 $d=9$ 时,NPE-SVM 软件缺陷预测各评价指标最优。

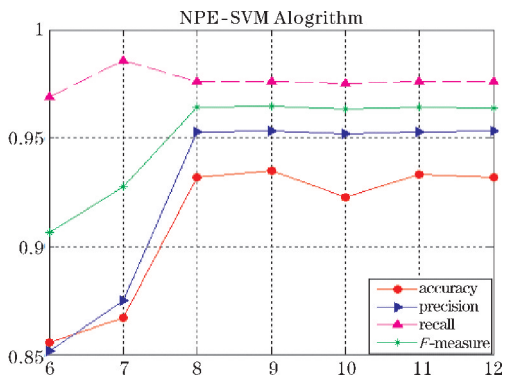

图 2 NPE 算法不同 *d* 值下预测结果

选用 NASA 中的 3 个数据集,即 PC1、CM1 和 PC4,分别包含 1107、505 和 3840 个模块。这些项目主要使用 C、C++等语言开发。从 3 个数据集中分别随机抽取 400 条作为实验数据。使用相同的实验数据,比较本文模型与单一的 SVM 及 LLE-SVM 模型的预测能力,实验结果分别如图表 2 ~4 所示。

表 2 PC1 数据集预测结果比较/%

算法	准确度	查准率	查全率	*F*
SVM	72.22	79.03	87.50	83.05
LLE-SVM	80.00	79.37	94.04	87.72
NPE-SVM	81.57	80.30	96.36	87.80

表 3 CM1 数据集预测结果比较/%

算法	准确度	查准率	查全率	*F*
SVM	81.56	84.42	91.22	91.50
LLE-SVM	82.22	86.21	94.12	92.89
NPE-SVM	93.33	95.35	97.62	96.47

表 4 PC4 数据集预测结果比较/%

算法	准确度	查准率	查全率	*F* 值
SVM	72.41	81.22	90.14	85.44
LLE-SVM	75.86	83.45	94.12	89.93
NPE-SVM	82.62	82.36	97.12	89.47

需要指出的是,不同参考文献中选取的数据集都具有随机性,选用不同数据集得到的测试结果存在一定的差异,所以本文的实验结果与参考文献[10]中有所不同。但本文实验是在相同数据集下比较各预测模型的评价指标,因此,结果对比是有效的。此外,软件缺陷数据集的类不均衡问题对预测结果也有所影响,如 CM1 数据集中共包含 3840 个软件模块,但仅 68 个模块包含缺陷,即使实验预测准确度较高,但 TP 和 TP+FP 所占比例仍然较小,这将导致实验的查准率(Precision)较低。以上实验结果表明,该软件缺陷预测模型在使用 3 个数据集实验中,4 个评价性能指标整体上明显优于其他模型。相比单一 SVM 预测模型,使用 LLE-SVM 预测模型,各指标有所提高。但 LLE-SVM 预测模型使用 LLE 算法对邻近样本数的选择敏感,不同的邻近数对最后的降维结果有很大的影响。而本文的算法得到了降维后更稳健、更准确的解,使得各评价指标提高了2% ~4%。

4　结论

软件缺陷预测是软件开发过程中一项重要技术,有效地预测软件缺陷能够合理地分配测有用资源[11-12]。文中基于流行学习中保留局部几何特性的 NPE 算法,针对软件缺陷数据集分配不均衡、高维、小采样及非线性问题,提出了基于 NPE-SVM 的软件缺陷预测模型。该模型的主要思想是将奇异的广义特征计算问题转化为两个特征分解问题,从而得到了更稳定的特征向量。实验结果表明,本文模型有明显的优势,但并未考虑各预测模型所花费的空间和时间成本。下一步,将上述条件考虑在内,对比各模型的预测结果,提出改进方案,减少预测所需的空间和时间成本。

参考文献:

[1] 何中威,范鑫. 软件缺陷预测技术研究[J]. 中国新通信,2015(22):127.

[2] 陈琳. 基于机器学习的软件缺陷预测研究[D]. 重庆:重庆大学,2016.

[3] Wang Y, Wu Y. Complete neighborhood preserving embedding for face recognition [J]. Pattern Recognition,2010,43(3):1008-1015.

[4] He P, Li B, Liu X, et al. An empirical study software defect prediction with a simplified metric set [J]. Information and Software Technology,2015,59:170-190.

[5] 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.

[6] 张强. 局部线性嵌入算法的改进及其在人脸识别中的应用[D]. 重庆:重庆理工大学,2017.

[7] 仝一君,王力. 基于 NPE 算法的语音特征提取应用研究[J]. 通信技术,2014,47(11):1281-1284.

[8] 胡迪(SALAHUDDIN). 软件缺陷预测算法研究[D]. 哈尔滨:哈尔滨工业大学,2017.

[9] 韦良芬. 基于机器学习的软件缺陷预测技术研究[J]. 长春大学学报,2017,27(10):13-15.

[10] 刘光永. 基于 LLE-SVM 软件缺陷预测模型的研究[D]. 天津:天津大学,2014.

[11] 甘露,臧洌,李航. 基于 DA-SVM 的软件缺陷预测模型[J]. 计算机与现代化,2017(2):36-39,44.

[12] Issam H. Laradji, Mohammad Alshayeb, Lahouari Ghouti. Software defect prediction using ensemble learning on selected features[J]. Information and Software Technology,2015(58):388-402.

Software Defect Prediction Model based on NPE-SVM

WANG Yu-hong[1], FAN Jing[1], LEI Min[2,3], SUN Hui-zhong[2]

(1. School of Electrical and Information Engineering, Yunnan University for Nationalities, Yunnan 650000, China; 2. Security Center, Beijing University of Posts and Telecommunications Information, Beijing 100876, China; 3. GuiZhou University, Guizhou Provincial Key Laboratory of Public Big Data, Guiyang 550025, China)

Abstract: Aiming at the problem of data set imbalance, high dimension, small sampling and nonlinear dimensionality reduction in software defect prediction, this paper proposes a software defect prediction model based on Neighborhood preserving embedding support vector machine . The model uses NPE algorithm to reduce the dimension of the dataset. By transforming the singular generalized feature in the NPE algorithm into two feature decomposition problems, get a more accurate and robust solution, which effectively avoids the prediction caused by the attribute reduction decrease in accuracy. The support vector machine is chosen as the basic classifier. The simulation results show that compared with other methods, the recalling rate and *F*-measure index of the evaluation model are significantly increased by 2% ~4%.

Keywords: software defects; neighborhood preserving embedding; machine learning; pattern recognition; manifold learning