基于改进 MDS 的软件缺陷预测

史雪静[1],吴 飞[2],荆晓远[1,3]
(1.南京邮电大学 计算机学院,江苏 南京 210003;
2.南京邮电大学 自动化学院,江苏 南京 210003;
3.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)

摘 要: 随着计算机技术的发展,计算机软件产品给个人和企业都带来了很多方便,但很多软件也会存在各种缺陷。为了找到并解决软件中存在的缺陷,研究者将机器学习等方法应用到软件缺陷预测之中,但这些方法在数据预处理方面还存在很多需要改善的地方。在之前的研究中,有研究者使用多维尺度分析(MDS)对数据样本进行降维,但关于如何使用和改善 MDS 的方法却很少。文中提出了基于阈值相关性的多维尺度分析(TC_MDS)方法,在使用 MDS 方法的基础上,使用对称不确定性(SU)方法提取具有高鉴别的特征,并使用阈值相关性去除冗余特征。该方法学习得到的数据具有高鉴别性,去除了冗余特征,从而提高了预测效率。在软件工程 NASA 数据库上的实验结果表明,提出的方法具有较好的缺陷预测效果。

关键词: 多维尺度分析; 对称不确定性; 阈值相关性; 软件缺陷预测
中图分类号: TP31 **文献标识码:** A **文章编号:** 1673-629X(2017)12-0020-03
doi: 10.3969/j.issn.1673-629X.2017.12.005

Software Defect Prediction Based on Improved MDS

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

Abstract: With the development of computer technology,computer software products have brought many convenience to individuals and businesses,but many software may have a variety of defects.In order to find and solve them,researchers have applied machine learning and other methods in software default prediction,but they need to be improved on data preprocessing.In previous studies,the researchers used Multi-Dimensional Scaling (MDS) to reduce the dimensionality of data samples.But the methods about how to use and improve MDS are few.A method of Threshold Correlation on MDS (TC_MDS) is proposed in this paper.Based on MDS,Symmetrical Uncertainty (SU) is used to extract the features with high discriminatory and threshold correlation to remove the redundancy.The method makes the data with high discriminatory,removing of redundancy,improvement of forecasting efficiency.The results on NASA database show it has very good defect prediction effect.

Key words: MDS; symmetrical uncertainty; threshold correlation; software defect prediction

0 引 言

软件缺陷预测可以预测软件出现的错误[1]。从整体上,软件缺陷预测可以分为动态缺陷预测和静态缺陷预测[2]。文中使用的是静态缺陷预测。

至今已有很多文献提出了静态软件缺陷预测算法,算法的核心有两点,一是挖掘软件度量,二是构建软件缺陷预测模型。目前已经有很多研究者将机器学习方法运用到软件缺陷预测中,例如 K 近邻分类器(K-Nearest Neighbor,KNN)、压缩 C4.5 模型(Compressed C4.5,CC4.5)[3]、朴素贝叶斯模型(Naïve Bayes,NB)[4]、支持向量机模型(Support Vector Machine,SVM)[5-6]、神经网络模型(Neural Networks,NN)[7-8]

收稿日期: 2017-01-20 **修回日期:** 2017-05-25 **网络出版时间:** 2017-09-27
基金项目: 国家自然科学基金资助项目(61272273)
作者简介: 史雪静(1991-),女,硕士研究生,研究方向为软件缺陷预测;吴 飞,讲师,研究方向为机器学习、软件工程;荆晓远,教授,博士生导师,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.20170927.1000.072.html

等。

软件度量即是与软件是否有缺陷密切相关的属性,FENTON 等[9] 将软件度量分为产品度量、过程度量、资源度量。这些度量能够描述一个软件模块的各种属性,预测模型就是在度量和缺陷类型上建立的关系,所以拥有高质量的软件度量尤为重要。主成分分析(Principal Component Analysis,PCA)[10]、线性鉴别分析(Linear Discriminant Analysis,LDA),拉普拉斯特征映射方法(Laplacian Eigenmaps,LE)[11]、多维尺度分析(Multi-Dimensional Scaling,MDS)[12-13] 等都是针对软件质量进行的研究。

在研究 MDS 方法的基础上,文中使用对称不确定性(Symmetrical Uncertainty,SU)方法选出高鉴别性的特征,并去除冗余特征,提出了一种新的数据预处理方法,即基于阈值相关性的多维尺度分析(Threshold Correlation on Multi-Dimensional Scaling,TC_MDS)。在 NASA 数据库[14] 上对该方法的有效性进行了验证。

1　多维尺度分析

MDS 是一种维度降低的方法,通过分析相似数据来挖掘数据中的隐藏结构信息。通常,相似度量使用欧氏距离表示。所以,MDS 算法的目的是在尽可能保留数据样本间距离的情况下,将数据样本映射到一个低维空间,以此降低样本的维度。

给定一组训练样本 $X=\{x_i,l_i\}$,$i=1,2,\cdots,n$,其中训练样本 $x_i\in R^d$,d 是样本维数,l_i($l_i\in\{1,2,\cdots,c\}$)是样本类别标签。两个样本之间距离由欧氏距离定义为:

$$d(x_i,x_j)=(x_i-x_j)^{\mathrm{T}}(x_i-x_j) \tag{1}$$

每对样本间的距离组成的矩阵作为相似矩阵。

$$\boldsymbol{D}=(d(x_i,x_j)^2)_{i,j=1}^{n} \tag{2}$$

MDS 的目标是给定 $\boldsymbol{D}$,构造样本 $Z=\{z_i,l_i\}$,$i=1,2,\cdots,n$,使得 $\|z_i-z_j\|\approx d_{i,j}$,其中 $i,j\in 1,2,\cdots,n$。可将 MDS 的解决看作一个优化问题,通过最小化下面的代价函数,可求得

$$\min_{z_1,\cdots,z_n}\sum_{i<j}(\|z_i-z_j\|-d_{i,j})^2 \tag{3}$$

记 n 维向量为 $\varphi=[x_1^{\mathrm{T}}x_1,x_2^{\mathrm{T}}x_2,\cdots,x_n^{\mathrm{T}}x_n]$,则 $\boldsymbol{D}=\varphi\boldsymbol{e}_n^{\mathrm{T}}-2\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}+\boldsymbol{e}_n\varphi^n$。其中,$\boldsymbol{e}_n$ 为数值均为 1 的 n 维列向量,那么存在矩阵 $\boldsymbol{T}$,满足:

$$\boldsymbol{T}=-\frac{(\boldsymbol{I}-\boldsymbol{e}_n\boldsymbol{e}_n^{\mathrm{T}})\boldsymbol{D}(\boldsymbol{I}-\boldsymbol{e}_n\boldsymbol{e}_n^{\mathrm{T}})}{2}=\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X} \tag{4}$$

通过广义特征值求解方法计算 $\boldsymbol{T}$ 的特征值和特征向量,取前 $m(m<d)$ 个较大特征值 $(\lambda_1,\lambda_2,\cdots,\lambda_m)$ 对应的特征向量,则所求低维样本为:

$$\boldsymbol{Z}=\mathrm{diag}(\lambda_1^{1/2},\lambda_2^{1/2},\cdots,\lambda_m^{1/2})\boldsymbol{V}^{\mathrm{T}} \tag{5}$$

2　基于阈值相关性的多维尺度分析

文中在 MDS 的基础上进行改进,引入方法 SU[15]。SU 作为非线性的相关性度量,使用的理论来自信息论中的熵。该方法可以用来评估特征的质量。对于两个变量 X 和 Y,SU 的计算公式如下:

$$\mathrm{SU}(X,Y)=2\times\frac{\mathrm{IG}(X\mid Y)}{H(X)+H(Y)} \tag{6}$$

其中,$H(X)$ 为变量 X 的熵。

假设 $p(x)$ 是 X 取值的概率,则 $H(X)$ 为:

$$H(X)=-\sum_{x\in X}p(x)\log_2 p(x) \tag{7}$$

$\mathrm{IG}(X\mid Y)$ 为信息增益,定义为:

$$\mathrm{IG}(X\mid Y)=H(X)-H(X\mid Y)=H(Y)-H(Y\mid X) \tag{8}$$

$\mathrm{SU}(X,Y)=1$ 意味着通过任意一个变量的取值能完全预测另一变量的值;$\mathrm{SU}(X,Y)=0$ 则表明 X 和 Y 之间相互独立。

虽然 MDS 方法获得的新样本在降低计算复杂度的同时包含了尽可能多的原样本信息,但是该方法对分类效果改善很少,通过使用 SU 方法选取与类别相关性较高的特征可以提高新样本的鉴别性。另外,MDS 无法去除冗余特征,即降维后的样本可能会含有冗余特征,这会降低算法的效率,所以文中使用阈值相关性方法去除冗余特征。

由此,TC_MDS 的实施步骤如下:

(1) 使用 MDS 对样本集降维;

(2) 使用 SU 方法计算每个特征与类别的相关度,提取具有高鉴别性的特征;

(3) 使用阈值相关性方法去除冗余特征。

在实际软件度量中,存在非线性的关系,所以文中依然选择 SU 来计算一对特征间的相似度。文中的阈值相关性方法使用预设的 β 作为相关性的临界值,在步骤(2)得到的特征下,从后向前对每个特征进行相关性分析,所有大于临界值的一对特征就从样本集中去除靠后的特征,然后以此类推。之所以从后向前进行相关性分析,是因为步骤(2)得到的特征从后往前其鉴别性越来越高,所以从后往前进行相关性分析,当遇到相关度大于 β 值的两个特征时,就可以优先去掉鉴别性小的特征,从而保留鉴别性较大的特征。

TC_MDS 算法描述如下:

输入:训练样本集 $X=[X_1,X_2,\cdots,X_c]$,其中 $X_i=(F_1,F_2,\cdots,F_m,L)$,$i=1,2,\cdots,c$,相关性阈值 β;

输出:样本集 Z。

步骤 1:计算各个样本之间的距离,得到距离矩阵 $\boldsymbol{D}$;

步骤 2:使用 MDS 降维,得到降维后样本集 $X'=$

$[X_1', X_2', \cdots, X_c']$，其中 $X_i' = (F_1, F_2, \cdots, F_k, L)$，$k < m$，$i = 1, 2, \cdots, c$；

步骤 3：令 $i = 1$ to k，循环

计算 $S_i = \mathrm{SU}(F_i, L)$；

步骤 4：对 S_i 按从大到小排序；

步骤 5：将序列中最前面的 g 个特征作为新样本的特征，得到样本集 $X'' = [X_1'', X_2'', \cdots, X_c'']$，其中 $X_i'' = (F_1, F_2, \cdots, F_g, L)$，$g < k$，$i = 1, 2, \cdots, c$；

步骤 6：对每对特征从后向前使用 SU 进行相关性分析，去除大于 β 的指定特征，得出最终样本 Z。

3 实 验

3.1 数据库

选用 NASA 数据库，五个工程分别代表着 NASA 的软件系统，它们具有不同的度量和对应的缺陷标记。表 1 汇总了这五个工程的详细信息。

表 1 NASA 数据集

数据集	缺陷样本数	样本总数	特征数	缺陷样本占比/%
CM1	42	344	37	12.21
MW1	27	255	37	10.59
PC1	61	711	37	8.58
PC3	134	1 079	37	12.42
PC4	177	1 288	37	13.74

3.2 性能评价指标

实验中使用四种评估指标，分别是召回率(Recall，Pd)、误检率(Pf)、F-measure 和 ROC 曲线下的面积(AUC)。预测结果见表 2。

表 2 四种预测结果

	Predict as defective	Predict as defect-free
Defective modules	A	B
Defective-free modules	C	D

指标定义为：

$$\mathrm{Pd} = A/(A+B) \tag{9}$$

$$\mathrm{Pf} = C/(C+D) \tag{10}$$

$$\text{F-measure} = 2*\ \text{recall}*\ \text{precision}/(\text{recall}+\text{precision}) \tag{11}$$

其中，$\mathrm{precision} = A/(A+C)$。

当具有较高的 Pd，F-measure，AUC 和较低的 Pf，一个预测模型才算是好的。且 F-measure 和 AUC 是综合性评价指标，更为重要。

3.3 实验结果与分析

实验使用 MDS、PCA、LE 作为对比方法，使用随机森林作为分类器，在 CM1、MW1、PC1、PC3、PC4 库中进行实验，结果见表 3。

表 3 实验结果

datasets	measure	LE	PCA	MDS	TC_MDS
CM1	Pd	0.44	0.26	0.15	0.66
	Pf	0.18	0.11	0.04	0.17
	F-measure	0.32	0.25	0.20	0.51
	AUC	0.52	0.43	0.45	0.72
MW1	Pd	0.49	0.29	0.21	0.61
	Pf	0.19	0.17	0.15	0.25
	F-measure	0.31	0.27	0.27	0.33
	AUC	0.41	0.43	0.42	0.52
PC1	Pd	0.66	0.38	0.54	0.70
	Pf	0.10	0.09	0.17	0.22
	F-measure	0.35	0.32	0.32	0.53
	AUC	0.59	0.50	0.51	0.76
PC3	Pd	0.65	0.34	0.64	0.65
	Pf	0.23	0.09	0.19	0.25
	F-measure	0.35	0.29	0.28	0.38
	AUC	0.59	0.52	0.53	0.62
PC4	Pd	0.62	0.39	0.28	0.49
	Pf	0.31	0.13	0.09	0.08
	F-measure	0.40	0.36	0.29	0.49
	AUC	0.49	0.45	0.45	0.54

分析表 3 可知，提出的方法 TC_MDS 在各个数据库上的缺陷预测效果普遍好于其他方法，尤其是 F-measure 和 AUC，对比 PCA、LE 以及 MDS 优势明显，说明了该方法在缺陷预测中的优势。

4 结束语

在 MDS 的基础上，使用 SU 方法提取有鉴别性的特征，并使用阈值相关性方法去除冗余特征，提出一种新的数据预处理方法(TC_MDS)。该方法学习得到的数据具有很好的鉴别性，并去除了冗余特征，提高了运算效率，降低了复杂度。在 NASA 数据库上的实验结果表明，TC_MDS 与现有的代表性缺陷预测方法相比，明显提高了缺陷预测的效率。

参考文献：

[1] 李 勇，黄志球，房丙午，等.代价敏感分类的软件缺陷预测方法[J].计算机科学与探索，2014，8(12)：1442-1451.

[2] 王 青，伍书剑，李明树.软件缺陷预测技术[J].软件学报，2008，19(7)：1565-1580.

[3] WANG J，SHEN B J，CHEN Y T.Compressed C4.5 models for software defect prediction [C]//International conference

(下转第 27 页)

络架构中控制平面所面临的压力。对学术界目前提出的 SDN 控制器解决方案的优缺点做了简要介绍。同时提出了一种 SDN 控制器的优化解决方案,即功能模块化的控制平面架构。阐明了 SDN 控制平面功能模块化的设计思路,其中包括功能模块化控制平面的整体架构,控制器内部各部分组成,控制器内部通信流程,以及采用这种架构下的控制器在接收不同消息时的处理流程。最后通过搭建拓扑环境对控制器进行测试。结果表明,在这种架构下的控制平面能够对网络进行有效的管控,同时功能模块的部署也更灵活,提高了控制器的可扩展性。

参考文献:

[1] Feamster N,Rexford J,Zegura E.The road to SDN: an intellectual history of programmable networks [J] .ACM SIGCOMM Computer Communication Review,2014,44(2):87-98.

[2] McKeown N,Anderson T,Balakrishnan H,et al.OpenFlow: enabling innovation in campus networks [J] .ACM SIGCOMM Computer Communication Review,2008,38(2):69-74.

[3] 张顺淼,邹复民.软件定义网络研究综述 [J] .计算机应用研究,2013,30(8):2246-2251.

[4] Gude N,Koponen T,Pettit J,et al.NOX: towards an operating system for networks [J] .ACM SIGCOMM Computer Communication Review,2008,38(3):105-110.

[5] 江国龙.SDN 控制器: Beacon 核心技术分析 [J] .程序员,2014(2):107-111.

[6] 李立龙,吕光宏,董永彬.基于 OpenFlow 的 SDN 控制平面可扩展性综述 [J] .电子科技,2015,28(1):171-175.

[7] 房秉毅,张　歌,张云勇,等.开源 SDN 控制器发展现状研究 [J] .邮电设计技术,2014(7):29-36.

[8] Dixit A,Hao F,Mukherjee S,et al.Towards an elastic distributed SDN controller [J] .ACM SIGCOMM Computer Communication Review,2013,43(4):7-12.

[9] Jimenez Y,Cervello-Pastor C,Garcia A J.On the controller placement for designing a distributed SDN control layer [C] //Networking conference. [s.l.]:IEEE,2014:1-9.

[10] Tootoonchian A,Ganjali Y.HyperFlow: a distributed control plane for OpenFlow [C] //Internet network management conference on research on enterprise networking. [s.l.]:USENIX Association,2010:3.

[11] Koponen T,Casado M,Gude N,et al.Onix: a distributed control platform for large-scale production networks [C] //USENIX symposium on operating systems design and implementation.Vancouver,BC,Canada:USENIX,2010:351-364.

[12] Yeganeh S H,Ganjali Y.Kandoo: a framework for efficient and scalable offloading of control applications [C] //Workshop on hot topics in software defined networks. [s.l.]:ACM,2012:19-24.

[13] 鲜永菊,朱　佳,鲁昭男.基于 SDN 的多控制器部署策略的研究 [J] .电视技术,2016,40(6):78-84.

[14] Lin P,Bi J,Chen Z,et al.WE-bridge: west-east bridge for SDN inter-domain network peering [C] //IEEE conference on computer communications workshops. [s.l.]:IEEE,2014:111-112.

[15] 李军飞,兰巨龙,胡宇翔,等.SDN 多控制器一致性的量化研究 [J] .通信学报,2016,37(6):86-93.

(上接第 22 页)

on quality software. [s.l.]:IEEE,2012.

[4] TAO W,LI W H.Naive Bayes software defect prediction model [C] //International conference on computational intelligence and software engineering.Wuhan,China:IEEE,2010:1-4.

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

[6] SHEPPERD M,BOWES D,HALL T.Researcher bias: the use of machine learning in software defect prediction [J] .IEEE Transactions on Software Engineering,2014,40(6):603-616.

[7] QUAH T S,THWIN M M T.Application of neural networks for software quality prediction using object-oriented metrics [C] //International conference on software maintenance. [s.l.]:IEEE,2003:589-590.

[8] ZHENG J.Cost-sensitive boosting neural networks for software defect prediction [J] .Expert Systems with Applications,2010,37(6):4537-4543.

[9] FENTON N E,NEIL M.Software metrics: roadmap [C] //Proceedings of conference on the future of software engineering. [s.l.]: [s.n.],2000:357-370.

[10] 刘　旸.基于机器学习的软件缺陷预测研究 [J] .计算机工程与应用,2006,42(28):49-53.

[11] BELKIN M,NIYOGI P.Laplacian eigenmaps for dimensionality reduction and data representation [J] .Neural Computation,2003,15(6):1373-1396.

[12] BRUCHER M,HEINRICH C,HEITZ F.A metric multidimensional scaling-based nonlinear manifold learning approach for unsupervised data reduction [J] .Eurasip Journal on Advances in Signal Processing,2008(1):1-12.

[13] LU H,CUKIC B,CULP M.A semi-supervised approach to software defect prediction [C] //Computer software and applications conference. [s.l.]:IEEE,2014:416-425.

[14] MENZIES T,GREENWALD J,FRANK A.Data mining static code attributes to learn defect predictors [J] .IEEE Transactions on Software Engineering,2007,33(1):2-13.

[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.