基于支持向量机的软件缺陷预测模型

王 涛，李伟华，刘 尊，史豪斌

（西北工业大学 计算机学院，陕西 西安 710072）

摘 要：软件缺陷预测在软件系统开发的各个阶段发挥着极为重要的作用。利用机器学习的相关方法建立更好的预测模型已经被广泛研究。文章分析了支持向量机 SVM 作为二值分类模型应用到软件缺陷预测中的实现方法，构造了基于 SVM 的可迭代增强的缺陷预测模型 SVM-DP。在 13 个基准数据集上开展比较实验，定量地分析了应用各种核函数对 SVM-DP 模型性能的影响。实验结果显示，应用线性内积核函数的 SVM-DP 具有最优的预测性能。同时，在与 J48 的比较实验中，最高超过 J48 预测模型 20% 的性能进一步证明了 SVM-DP 模型应用于软件缺陷预测的有效性。

关 键 词：软件缺陷预测，软件度量，支持向量机
中图分类号：TP311.5 **文献标识码**：A **文章编号**：1000-2758(2011)06-0864-07

软件测试是一项极其耗费成本的工作。通常认为缺陷在软件系统中的分布是不均匀的，20% 的模块包含超过 80% 的缺陷。因此，软件缺陷预测的目的，是在软件系统开发生命周期的各个阶段鉴别可能存在缺陷的模块，为测试人员提供缺陷分布等信息，以此指导软件质量确保工作，这一流程已经成为事实上的工业标准[1]。机器学习的方法将缺陷预测看作二值分类问题。模型构造完成后，对分析目标（可能是文件、模块、方法等）进行分类预测，从而在一定条件下做出判决。B. Turhan 等人[2]提出了一种多变量方法与贝叶斯理论结合的预测模型，并且强调了利用特征选取技术对静态代码属性进行筛选的重要性。实验结果显示，其预测模型有很好的性能。T. Gyimothy 等人[3]分析并使用了逻辑回归、线性回归、决策树及神经网络等四种方法构造预测模型，并且试图找出最优的代码静态属性。在二值分类方面支持向量机 SVM 一直被广泛采用[4]，SVM 是建立在统计学习理论的 VC 维理论和结构风险最小原理基础上的，根据有限的样本信息在模型的复杂性和学习能力之间寻求平衡，以期获得最好的泛化能力[5]。本文分析了利用 SVM 构造缺陷预测模型的理论方法，构造了基于 SVM 的预测模型 SVM-DP。分析了利用不同核函数对 SVM-DP 性能的影响，并且与已知的其它方法进行了比较。

1 可迭代增强的 SVM 缺陷预测模型

研究人员普遍认为，软件的内部属性（如静态代码特征）和其外在表现（如缺陷）有联系。开发者收集同一个项目或本公司其它项目以及其它公司类似项目的历史数据，然后从其中抽取出代码的静态属性（通常利用软件度量元表示）。许多缺陷预测模型都是基于软件度量元提出的[2,3,6]。典型的静态代码属性包括 LOC、McCabe 以及 Halstead 等，被广泛用于抽象源码的表征。本文基于 SVM 理论构造了一个可迭代增强的缺陷预测模型 SVM-DP（SVM Defect Prediction Model）。SVM-DP 将缺陷预测看作一个二值分类问题。首先从同一项目以往版本积累的历史数据中，抽取出静态代码属性再加上代码单元是否包含缺陷的类型标记，就完成了对代码的抽象表征，并以此 n（n = 代码属性个数 + 1）元组构成的向量作为学习型预测模型的训练样本。对预测学习器进行训练并生成预测模型 SVM-DP。接

收稿日期：2011-04-12 基金项目：国家自然科学基金（F020510）、西北工业大学博士论文创新基金（CX200815）及陕西省自然科学基础研究计划（2010JM8039）资助
作者简介：王 涛（1980—），西北工业大学博士研究生，主要从事信息安全、计算机网络及机器学习的研究。

着,对正在开发项目的当前版本进行预测,以期发现包含潜在缺陷的代码单元,报告缺陷分布信息甚至预测修补缺陷的工作量,从而指导软件测试人员将有限的成本资源集中到最需要的地方。预测得到的缺陷信息可以追加到历史数据中,形成增量式的历史数据集合。当进入项目新的生命周期后,由增量的历史数据集中抽取代码属性并训练生成增强的预测模型 advanced SVM-DP,进一步完成对新版本项目的预测工作。图 1 形象地描述了 SVM-DP 模型的迭代增量过程。

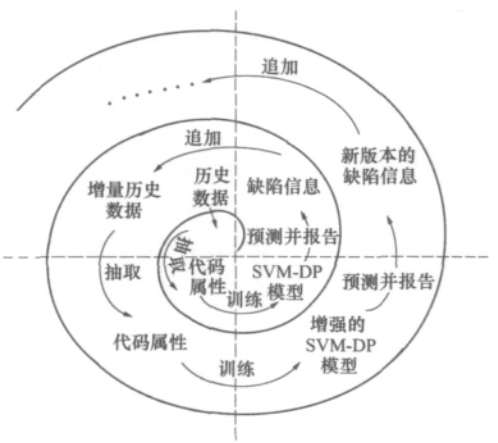

图 1　SVM-DP 模型的迭代增量过程

依据支持向量机的理论,例如在二维平面中假定 C_1 和 C_2 是要区分的两个类别,它们的样本如图 2 所示。中间的直线 S 就是一个分类函数,它可以将两类样本完全分开。这样引申出两个关键问题,首先,若样本线性不可分,则需要向高维空间转化,使其在高维空间中线性可分,因此将线性不可分的样本转化为线性可分是 SVM 中的一个关键问题。在不考虑空间维度的情况下,将分类函数统称为“超平面”。其次,将图 2 中的直线 S 适度旋转或平移仍然可将 C_1 和 C_2 分开,亦即存在多条直线将 C_1 和 C_2 分开。因此,如何找到最优分类超平面是 SVM 中的又一个关键问题。

本文以软件模块为单元进行分析,软件模块是指代码中功能相对独立的一个部分,诸如 C 程序中的一个函数,JAVA 程序中的一个方法。假定两个类别表示为 $c_i \in \{-1,1\}$,以及 N 个已经包含类别标注的训练样本: (m_1,c_1),$(m_2,c_2)\cdots,(m_N,c_N)$,$m_i \in \mathrm{R}^d$。其中,$d$ 为向量的维度。

如果两类问题线性可分,则存在理想的权重向量 w 使得 $\|w\|^2$ 最小,并且有

$$c_i(w \cdot m_i - b) \geqslant 1 \quad i = 1,2,\cdots,N \tag{1}$$

其中,使上式等号成立的样本称为支持(撑)向量。由支持向量可得到两个超平面将两类样本分割开来。如图 1 所示,两个超平面 S_1 和 S_2 之间的距离称为分类间隔(Margin),而当 $\frac{1}{2}\|w\|^2$ 最小时,Margin 最大,此时的两个超平面称为最优超平面。为求解 $\|w\|^2$ 最小,根据 Lagrange 乘数法,可引入 Lagrange 函数

$$L(w,b,\alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{N}\alpha_i(c_i(w \cdot m_i - b) - 1) \quad \alpha_i \geqslant 0 \tag{2}$$

求上式的极小值,得到 $w = \sum_{i=1}^{r}\alpha_i c_i m_i$,$r$ 为支持向量的个数。m_i 为训练向量,当 $\alpha_i > 0$ 时,对应的 m_i 即为支持向量。对于未知类别模块向量 m_i 可通过如下函数进行分类

$$F(m_i) = \mathrm{sgn}\{w \cdot m_i - b\} \tag{3}$$

式中,sgn 为符号函数。

当训练集非完全线性可分时,任何分类超平面对训练样本必有错划,此时无法达到所有样本点都满足约束条件即公式(1)。为此,可引入松弛变量 ξ_i 和惩罚因子 C,将原始最优化问题转化为求解如下最优化问题

$$\min_{w,b,\xi}\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i$$
$$\text{s.t. } c_i(w \cdot m_i - b) + \xi_i \geqslant 1$$
$$\xi_i \geqslant 0 \quad i = 1,2,\cdots,N \tag{4}$$

求得最优解 w^*、b^* 和 ξ^* 后,再构造分类超平面,进而求得新的决策函数

$$F'(m_i) = \mathrm{sgn}\{w^* \cdot m_i - b^*\} \tag{5}$$

对于非完全线性可分问题,也可引入核函数将

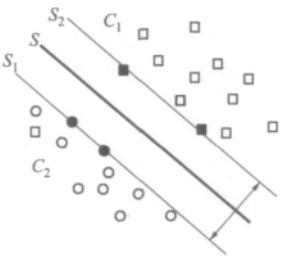

图 2　二维平面中的线性可分问题

非线性可分输入空间等价映射到线性可分输入空间。常见的核函数有:线性内积核函数(Linear)、径向基内积核函数(RBF)、多项式核函数(Polynomial)以及采用神经元的非线性作用函数Sigmoid作为内积的核函数等[4]。公式如下:

Linear 线性内积核函数

$$K(m_i,m_j) = (m_i \cdot m_j) + \gamma \quad (6)$$

RBF 核函数

$$K(m_i,m_j) = \exp(-\gamma \| m_i,m_j \|^2),\gamma > 0 \quad (7)$$

Polynomial 核函数

$$K(m_i,m_j) = [m_i \cdot m_j) + \gamma]^d \quad (8)$$

Sigmoid 核函数

$$K(m_i,m_j) = \tanh((m_i \cdot m_j) + \gamma) \quad (9)$$

表1 NASA MDP 数据集的基本信息

	CM1	JM1	KC1	KC3	KC4	MC1	MC2	MW1	PC1	PC2	PC3	PC4	PC5
开发语言	C	C	C++	Java	Perl	C++	C	C	C	C	C	C	C++
代码行数	20k	315k	43k	18k	25k	63k	6k	8k	40k	26k	40k	36k	164k
模块个数	505	10878	2107	458	125	9466	161	403	1107	5589	1563	1458	17186
包含缺陷模块个数	48	2102	325	43	75	68	52	31	76	23	160	178	516

此时,新的决策函数就成为

$$F''(m_i) = \mathrm{sgn}\{\sum_{i=1}^{r} \alpha_i c_i K(m_i,m_j) - b\} \quad (10)$$

2 数据集和代码属性

本文采用由NASA公布的NASA IV&V Facility Metrics Data Program(MDP)数据集[7],如表1所示,MDP包括13个不同的数据集,这些数据均来自NASA的实际软件项目,由最常见的开发语言编写。每个数据集包含来自不同软件项目的若干模块,规模从125/6k到17186/315k(模块个数/代码行)不等。与许多研究者采用私有软件项目数据不同,MDP是一个开放的数据仓库,数据集的开放性方便了不同的研究者进行实验的重复、改进甚至反驳,本文选取MDP的目的也在于此。

MDP数据集包含LOC、McCabe、Halstead以及其它等四类代码属性。所有类型的代码属性再加上是否包含缺陷的类型标记(defective)一共43个度量元,MDP数据集所提供的记录,一行代表一个模块,每个字段对应一个属性。依据SVM,每一记录构成的向量即为一个样本。

3 评价标准

为了构造基于SVM的最优预测模型以及在不同研究者之间进行性能比较,定义统一的评价标准非常有必要。然而,评价标准能否准确地反映预测模型的预测能力才是关键。依据数据挖掘理论的一般方法,为了定义评价标准首先要用到混淆矩阵(如表2所示)。

表2 二值分类的混淆矩阵

被分类为	True	False
确为True	TP	FN
确为False	FP	TN

在一个两类混淆矩阵中,实际为正类,预测也为正类的样本数量称为正确正类TP(true positive);实际为正类,预测为反类的称为错误反类FN(false negative);实际为反类,预测为正类的称为错误正类FP(false positive);实际为反类,预测也为反类的称为正确反类TN(true negative)。几个常用的评价标准定义如表3所示。

表3 常用评价标准

名称	计算公式
true positive rate, tpr	TP/(TP+FN)
Recall, R	TP/(TP+FN)
false positive rate, fpr	FP/(FP+TN)
Precision, P	TP/(TP+FP)
F-Measure, F	2* R* P/(R+P)

另一组评价指标是ROC曲线,以及AUC(area under curve)[8]。ROC曲线的X轴表示误报率fpr,Y轴表示召回率R亦即tpr。AUC是指ROC曲线下面包括的面积,即ROC曲线的积分。ROC曲线和AUC能够更准确地描述预测模型的性能,尤其在类分布密度不平衡以及各类误判代价不对称的情况。ROC曲线上的一组点是通过调整分类器决策阈值得到的,从而也避免了应用具体策略定义阈值给比

较研究带来的不便。ROC 曲线越靠近左上方,表示对应的分类器一般化能力越强,AUC 能以定量的方式表示该 ROC 曲线对应的分类器的一般化能力。

4　实验方法及结果分析

本文在 JAVA 环境中设计并开发了 SVM-DP 的原型系统,并使用了 LibSVM 3.0 工具包[9]。图 3 为实验算法的伪代码描述。对所有实验集的属性值的预处理包括: 删除原始数据集中的 Module 属性,该属性是 MDP 为数据集中模块样本的编号,与缺陷预测不相关; 依据 MDP 提供的说明,按 error_count 属性的值添加 defective 类标记属性; 对个别数据集中缺失的属性值进行补充,如 error_density 属性,按照 MDP 的相关说明文档,error_density = 1000 × (error_count/loc_total); 所有属性值除了 defective 以外均进行正则化处理。libsvm 各项参数选取默认值[9](在 Poly 核函数中,取 $d=3$),进行 10-folder 交叉验证实验。

```
1   Exp_1(trainSet,testSet){ //实验1,比较各种核函数)
2         SVM(trainSet,testSet,'Linear'};
3         SVM(trainSet,testSet,'Poly'};
4         SVM(trainSet,testSet,'RBF'};
5         SVM(trainSet,testSet,'Sigmoid'};
6         return result;
7   }
8   Exp_2(trainSet,testSet){ //实验2,SVM-DF vs. J48
9         SVM-DP    -1i(trainSet,testSet);
10        J48 (trainSet,testSet);
11        return result;
12  }
13  Data sets = {CM1,JM1,KC1,KC1,KC3,KC4,MC1,MC2,
14            MW1,PC1,PC2,PC3,PC4,PC5}; //数据集
15  for each D in Date sets
16      for m = 1 to 10
17        for n = 1 to 10//10* 10 的交叉实验
18        {
19            Dtrain = 90% of D; //训练集
20            Dtest = 10% of D; //测试集
21            Exp_1(Dtrain,Dtest);
22            Exp_2(Dtrain,Dtest);
23        }
```

图 3　实验算法

实验 1　不同的 SVM 实现方法对预测性能的影响

本文定量地分析了不同 SVM 核函数对 SVM-DP 预测性能的影响。在实验之初我们发现,线性内积核函数的 SVM(li-SVM)在很多数据集上(诸如 CM1、MC1 等)的预测准确率达到 100% 的同时,还能保持误报率为 0。其它几种不同核函数的 SVM 也能够在保持误报率为 0 的前提下,预测准确率均能达到 95% 以上。经过分析,这种令人惊讶的结果是由各数据集中 error_count 属性和 error_density 属性造成的(包括 KC1 中的 Error_report_in_1_yr 等三个相关属性)。原因在于,样本的类别属性 defective 的属性值来源就是 error_count,即 error_count = 0 则 defective = false,否则 defective = true。所以,error_count 等相关属性存在于样本向量中,预测时,预测模型只需依据该属性就能直接确定该样本的类型,从而造成了预测模型的无效。这些属性值对预测模型的影响未在采用 MDP 数据集的相关研究中报道[2,4]。因此,必须将 error_count 属性和 error_density 属性以及 KC1 中的 Error_report_in_1_yr 等三个相关属性从各数据集中删除。各预测模型在 MDP 上的预测性能如表 4 所示。实验结果显示多项式核函数 SVM(Poly-SVM)虽然保持了零误判率,但在大多数数据集上不具备召回的(识别包含缺陷模块)能力,即 Poly-SVM 将所有数据集的绝大部分样本都分类到了反类(false 即 defect-free),并且未将任何一个反类误判为正类(true 即 defective),Poly-SVM 最高只在 JM1、KC1 及 KC4 上具有不超过 30% 的召回率,不具备实践应用的能力; Sigmoid 核函数 SVM(Sig-SVM)在各数据集上表现出了较为均衡的准确率,除了 MC1、MC2 及 PC5 等以外,对其余数据集的召回率均在 45% 以上,其中在 PC4 上 Sig-SVM 达到了 75.1% 的召回率,以及 0.729 的 f-measure 值,表现出了在所有数据集上的性能峰值; 径向基内积核函数 SVM(RBF-SVM)在除 JM1、KC3、MC2、PC3 及 PC4 以外的多数数据集上的性能明显优于 Sig-SVM,其中在 MW1 上 RBF-SVM 达到了 97.6% 的准确率以及 0.837 的 f-measure 值,表现出了在所有数据集上的性能峰值,同时,RBF-SVM 在 KC3、MC2 及 PC5 上的召回率及 f-measure 值均较低,从而说明 RBF-SVM 的预测性能还不够稳定; li-SVM 在各数据集上的召回率均达到了 50% 以上,分类准确率达到 62% 以上,其中在 MW1 上 li-SVM 达

到了93.5%的召回率、零误判率和0.966的f-measure值，表现出了在所有数据集上的性能峰值，同时li-SVM是各预测模型中性能最稳定的。各数据集的平均值说明，li-SVM仅在误判率一项略低于RBF-SVM，其余各指标均为各预测模型之最高。在MDP数据集上71.2%的平均召回率以及79%的分类准确率证明了将SVM-DP(li-SVM)应用于缺陷预测的有效性。

表4 在MDP上不同SVM核函数对预测性能的影响

	Linear				Polynomial				RBF				Sigmoid			
	tpr	*fpr*	*P*	*F*	*tpr*	*fpr*	*P*	*F*	*tpr*	*fpr*	*P*	*F*	*tpr*	*fpr*	*P*	*F*
CM1	0.705	0.129	0.845	0.769	0	0	0	0	0.541	0.219	0.712	0.615	0.452	0.119	0.792	0.575
JM1	0.625	0.28	0.691	0.656	0.125	0	1	0.222	0.524	0.216	0.708	0.602	0.642	0.259	0.713	0.675
KC1	0.784	0.345	0.694	0.736	0.298	0	1	0.46	0.646	0.312	0.674	0.660	0.464	0.302	0.606	0.525
KC3	0.684	0.091	0.883	0.771	0	0	0	0	0.423	0.138	0.754	0.542	0.521	0.126	0.805	0.633
KC4	0.77	0.063	0.924	0.840	0.148	0	1	0.257	0.623	0.164	0.826	0.710	0.61	0.214	0.740	0.669
MC1	0.653	0.221	0.747	0.697	0.015	0	1	0.029	0.515	0.264	0.661	0.579	0.24	0.312	0.435	0.309
MC2	0.5	0.028	0.947	0.654	0.058	0	1	0.109	0.115	0.024	0.75	0.199	0.248	0.101	0.711	0.368
MW1	0.935	0	1.000	0.966	0	0	0	0	0.732	0.018	0.976	0.837	0.624	0.12	0.839	0.716
PC1	0.737	0.195	0.791	0.763	0	0	0	0	0.631	0.162	0.796	0.704	0.581	0.201	0.743	0.652
PC2	0.694	0.347	0.667	0.680	0	0	0	0	0.512	0.248	0.000	0.000	0.498	0.257	0.660	0.568
PC3	0.782	0.298	0.724	0.752	0.013	0	1	0.025	0.643	0.257	0.714	0.677	0.672	0.367	0.647	0.659
PC4	0.871	0.321	0.731	0.795	0.006	0	1	0..011	0.736	0.354	0.675	0.704	0.751	0.309	0.708	0.729
PC5	0.514	0.314	0.621	0.562	0.023	0	0.857	0.045	0.489	0.229	0.862	0.624	0.381	0.227	0.627	0.474
Mean	0.712	0.202	0.790	0.742	0.053	0.000	0.604	0.096	0.548	0.200	0.701	0.573	0.514	0.224	0.694	0.581

实验2 SVM-DP(li-SVM)与J48的比较

本文在定量分析了最优SVM预测模型之后，将SVM-DP(li-SVM)与缺陷预测中常用的决策树方法J48[5,8]进行比较，以进一步确定SVM-DP在缺陷预测中的有效性。由于数据量较大，本实验只选取不同开发语言的数据集MW1、MC1、KC3、KC4等四个数据集进行比较实验，J48预测模型利用Weka[10]工具实现。图4至图7分别显示在MW1、MC1、KC3、KC4四个数据集上，SVM-DP(li-SVM)与J48的预测性能比较。

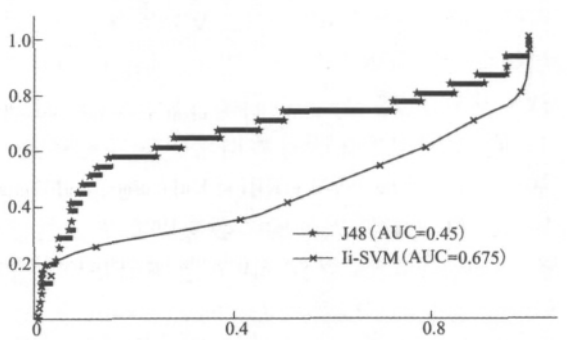

图4 SVM-DP(li-SVM)和J48在MW1上的ROC

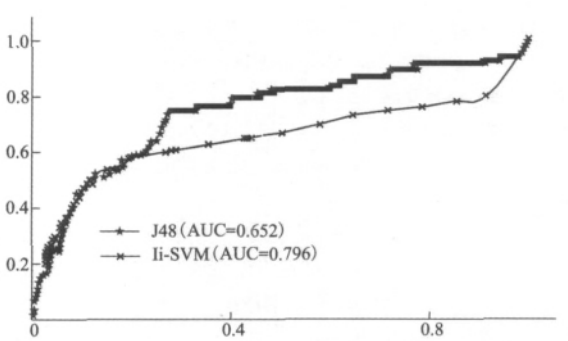

图5 SVM-DP(li-SVM)和J48在MC1上的ROC

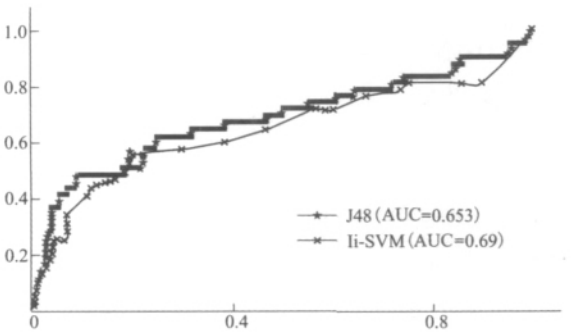

图 6　SVM-DP(li-SVM)和 J48 在 KC3 上的 ROC

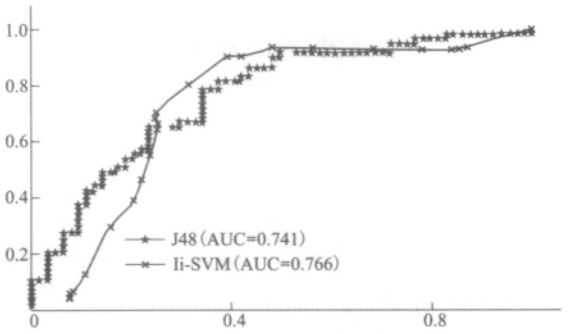

图 7　SVM-DP(li-SVM)和 J48 在 KC4 上的 ROC

从实验结果可以看出,SVM-DP(li-SVM)虽然在 KC3 和 KC4 上与 J48 相比性能接近,但比较全部四个数据集上的 ROC,SVM-DP(li-SVM)的预测性能优势明显。其中在 MW1(c)和 MC1(c++)数据集上,SVM-DP(li-SVM)的预测性能指标 AUC 分别为 0.675 和 0.796,比 J48 的预测性能明显高出 20%左右。

5　结　论

本文将支持向量机 SVM 引入软件缺陷预测中,分析了利用 SVM 构造预测模型的方法和过程。构造了可迭代增强的预测模型 SVM-DP。详细分析了软件缺陷预测领域公用的基准数据集 MDP,并针对实验对 MDP 进行了处理。在全部 13 个数据集上的比较实验证明,线性内积核函数 li-SVM 在平均召回率和平均分类准确率方面都优于其它类型的核函数。在分别由 C、C++、JAVA 以及 Perl 开发的 MW1、MC1、KC3、KC4 四个数据集上 SVM-DP(li-SVM)比缺陷预测中常用的 J48 表现出更优秀的预测性能。本文的研究工作显示了支持向量机在缺陷预测中的有效性,同时为在基准数据集上开展相关的比较研究提供了基础。

参考文献:

[1] Weyuker E J, Ostrand T J, Bell R M. Do Too Many Cooks Spoil the Broth? Using the Number of Developers to Enhance Defect Prediction Models. Empirical Software Engineering, 2008, 13(5): 539 ~ 559

[2] Turhan B, Bener A. A Multivariate Analysis of Static Code Attributes for Defect Prediction. Seventh International Conference on Quality Software, 2007, 231 ~ 237

[3] Gyimothy T, Ference R, Siket L. Empirical Validation of Object-Oriented Metrics on Open Source Software for Fault Prediction. IEEE Trans on Software Engineering, 2005, 31(10): 897 ~ 910

[4] Scholkopf B, Smola A J. Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Cambridge, MA, 2002

[5] Vapnik V, Golowich S, Smola A. Support Vector Method for Function Approximation, Regression Estimation, and Signal Processing. Mozer M, Jordan M, Petsche T (eds). Neural Information Processing Systems, MIT Press, 1997, 9

[6] Hongyu Zhang, Adam Nelson, Tim Menzies. On the Value of Learning from Defect Dense Components for Software Defect Prediction. In PROMISE10, Sep12-13, 2010. Timisoara, Romania

[7] Chapman M, Callis P, Jackson W. Metrics Data Program. NASA IV and V Facility, http://mdp.ivv.nasa.gov/, 2004

[8] Provost F, Fawcett T. Robust Classification for Imprecise Environments. Machine Learning, 2001, 42(3): 203 ~ 23

[9] Chang Chihchung, Lin Chihjen. LIBSVM: a Library for Support Vector Machines. Software available at http://www.csie.ntu.edu.tw/ ~ cjlin/libsvm

[10] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann. WEKA Data Mining Software: An Update: SIGKDD Explorations, Ian H. Witten, 2009, 11(1): 10 ~ 18

A Software DP (Defects Prediction) Model Based on SVM (Support Vector Machine)

Wang Tao, Li Weihua, Liu Zun, Shi Haobin

(Department of Computer Science and Engineering, Northwestern Polytechnical University, Xi'an 710072, China)

Abstract: Software defects prediction can help raise the effectiveness and efficiency of testing activities by constructing predictive classification models from static code attributes which can identify software modules with a higher than usual probability of defects. Our aim is to find the best performance predictive classification model through introducing SVM into DP. Sections 1 through 4 of the full paper explain our SVM-DP model and its application to analyzing the 13 data sets of NASA Metrics Data Program (MDP). Sections 1 through 4 are entitled: Iterative and Incremental Prediction Model SVM-DP (section 1); Benchmarking Data Sets and Code Metrics (section 2); Effectiveness Indicators (section 3); Experimental Method and Analysis of Test Results (section 4). Experimental results, presented in Table 4 and Figs. 4 through 7, and their analysis, show preliminarily the effectiveness of our SVM-DP model.

Key words: analysis, classification (of information), codes (symbols), data mining, defects, efficiency, evaluation, experiments, functions, iterative methods, maintenance, models, probability, software engineering; defects prediction (DP), software metrics, support vector machine (SVM)

2011版《挑大学 选专业》列出35校办光信息科学与技术本科专业在B等以上

武书连主编的《挑大学 选专业》2011版上有116校办理学电子信息科学类光信息科学与技术本科专业;A++等有3校,A+等有2校,A等有7校,B+等有12校,B等有11校。

A++等3校列第1至第3名,A+等2校列第4名及第5名,A等7校列第7名至第12名。A等以上共12校,依次为:华中科技大学、哈尔滨工业大学、复旦大学、中山大学、南开大学、北京交通大学、苏州大学、华南师范大学、电子科技大学、西安电子科技大学、山东大学、四川大学。

B+等以下,《挑大学 选专业》2011版不排办专业学校名次,这里在每校后括弧内注明该校总得分的排名。

B+等12校,按每校总得分排名依次为:上海交通大学(4),中国科学技术大学(16),东南大学(19),大连理工大学(21),北京理工大学(30),南京理工大学(47),山西大学(68),河北大学(78),浙江师范大学(100),福建师范大学(102),上海理工大学(118),长春理工大学(197)。

B等11校,按每校总得分排名依次为:武汉大学(8),吉林大学(12),西安交通大学(14),同济大学(25),西北工业大学(28),南京航空航天大学(37),西北大学(61),北京邮电大学(87),深圳大学(126),曲阜师范大学(171),中国计量学院(261)。

胡沛泉

2011年12月