文章编号: 1672-1497(2017)05-0086-05

基于支持向量回归的软件缺陷密度预测模型

杨腾翔[1]，万 琳[1]，王钦钊[2]，马振宇[1,3]，韩志贺[4]

(1. 陆军装甲兵学院信息工程系，北京 100072; 2. 陆军装甲兵学院控制工程系，北京 100072;
3. 陆军装甲兵学院技术保障工程系，北京 100072; 4. 31689 部队，吉林 四平 136000)

摘 要: 针对软件缺陷密度的预测问题，构建了一种基于支持向量回归(Support Vector Regression，SVR)的软件缺陷密度预测模型，指出影响回归预测精度的主要因素。首先，对软件度量元数据进行提取，利用归一化和随机序列的方法对缺陷数据进行预处理，并将数据分成训练集和测试集进行回归预测; 然后，引入网格搜索的方法对支持向量回归模型中的参数进行优化，大大提高了预测的精度; 最后，通过实验对比其他 5 种机器学习算法，验证了预测模型的有效性。

关键词: 支持向量回归(SVR)；网格搜索；软件缺陷密度预测

中图分类号: TP311.5 **文献标志码:** A **DOI:** 10.3969/j.issn.1672-1497.2017.05.017

Prediction Model of Software Defect Density Based on Support Vector Regression

YANG Teng-xiang[1], WAN Lin[1], WANG Qin-zhao[2], MA Zhen-yu[1,3], HAN Zhi-he[4]

(1. Department of Information Engineering, Army Academy of Armored Forces, Beijing 100072, China;
2. Department of Control Engineering, Army Academy of Armored Forces, Beijing 100072, China;
3. Department of Technical Support Engineering, Army Academy of Armored Forces, Beijing 100072, China;
4. Troop No. 31689 of PLA, Siping 136000, China)

Abstract: In order to predict the software defect density, a prediction model of software defect density based on Support Vector Regression (SVR) is constructed. The main factors that affect SVR prediction accuracy are data processing and parameter selection. Firstly, the metric metadata of software is extracted, and the normalization and the random sequences methods are used to preprocess the defect data. The data is divided into training set and test set for regression prediction. Then the grid search method is used to optimize the parameters of the SVR model, greatly improving the prediction accuracy. Finally, the validity of the prediction model is verified by comparing experiments with the other five kinds of machine learning algorithms.

Keywords: Support Vector Regression (SVR); grid search; software defect density prediction

软件缺陷预测主要用于判别软件中是否包含缺陷，预测软件中包含缺陷的数量或密度。预测过程属于机器学习的范畴，依据不同方法构建预测模型，完成软件缺陷的预测工作。软件缺陷预测[1-4]大致可分为静态缺陷预测和动态缺陷预测，其中: 静态缺陷预测主要是依据软件静态属性预测软件中是否包含缺陷; 动态缺陷预测则是为了建立缺陷与相应时间阶段之间的规则联系，记录出现缺陷数量较多的时间段。目前，软件缺陷预测模型主要有基于支持向量机(Support Vector Machine，SVM)的预测模型、基于神经网络的预测模型[5]和基于 LASSO-SVM 的预测模型[6]等。另外，研究者结合数据挖掘算法对模型的预测效果不断进行了改进，如: 姜慧研等[7]提出了利用蚁群算法对软件缺陷预测模型进行改

收稿日期: 2017-06-02
作者简介: 杨腾翔(1993-)，男，硕士研究生。

进；王男帅等[8]提出了利用遗传算法优化缺陷数据属性和 SVM 参数的预测模型。

现阶段，虽然缺陷预测技术一定程度上取得了相应进展，但仍存在以下问题：1）度量元的选择并没有得出系统的论证，缺陷的度量元有几十种且相互关联，在预测过程中需选择对缺陷预测影响较大的度量元；2）数据集对现有缺陷预测模型的影响较大，对一组数据预测结果较好的模型，换一组数据或许就达不到预测的效果；3）缺陷数据的预处理和模型参数的选择对预测结果的影响也较大。

针对上述问题，笔者提出一种基于支持向量回归（Support Vector Regression，SVR）的软件缺陷密度预测模型。首先，考虑不同度量元对缺陷密度预测的影响，选择与缺陷密度相关性较大的度量元，利用 SVR 构建缺陷密度预测模型，并对预测数据进行归一化和随机排序，以降低数据集对预测模型的影响；然后，选取径向基核函数（Radial Basis Function，RBF），并利用网格搜索的方法对模型参数进行优化；最后，通过实验验证预测模型的有效性。

1　软件缺陷密度预测方法框架

王青等[9]研究表明：缺陷的存在符合二八原则，即 80% 的缺陷往往存在于 20% 的程序模块中。缺陷密度是对软件、程序模块或源文件中缺陷数量和代码行数关系的度量，反映了该模块包含缺陷的概率。度量元是对程序内在属性的一种直观表示，反映了程序代码的质量和复杂度。笔者将度量元、缺陷密度分别作为预测过程的输入、输出，通过预测模型建立二者之间的关系，用于下一步的预测。图 1 为软件项目的缺陷密度预测模型。

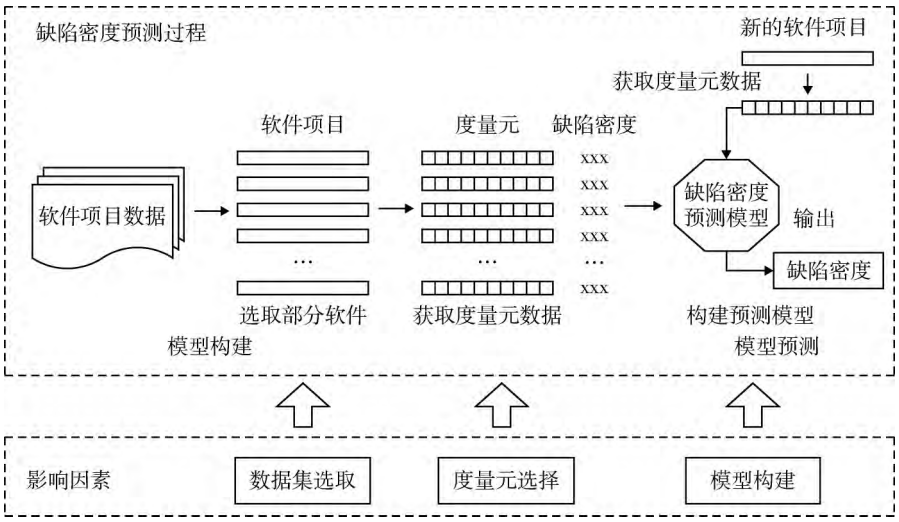

图 1　软件项目的缺陷密度预测模型

2　软件缺陷密度预测模型

2.1　度量元数据提取

本文引入 Spearman 秩相关系数的分析方法，用于衡量度量元和缺陷密度之间的关系，从而提取对缺陷密度影响较大的度量元，去除冗余的度量元。

给定软件项目某度量元 X 和缺陷密度 Y 的 n 组数据，可表示为 $X(X_1,\cdots,X_k,\cdots,X_n)$ 和 $Y(y_1,\cdots,y_k,\cdots,y_n)$。其中：$X_k$ 为第 k 个软件项目的度量元；y_k 为第 k 个软件项目的缺陷密度。

为分析度量元 X 和缺陷密度 Y 之间的相关程度，首先将 X 和 Y 中数据进行配对，得到数据组 $(X_1,y_1),\cdots,(X_k,y_k),\cdots,(X_n,y_n)$；然后将 X_k 和 y_k 分别按照大小排序，获得 X_k 和 y_k 在 2 个顺序样本中的排名（即秩），记作 R_k 和 S_k。由此可以得到数据组的 n 对秩，即 $(R_1,S_1),\cdots,(R_k,S_k),\cdots,(R_n,S_n)$。利用 Spearman 秩相关系数衡量 X 和 Y 之间的相关程度，可表示为

$$r=\frac{\sum_{k=1}^{n}(R_k-\bar{R})(S_k-\bar{S})}{\sqrt{\sum_{k=1}^{n}(R_k-\bar{R})^2\sum_{k=1}^{n}(S_k-\bar{S})^2}}=1-\frac{6\sum_{k=1}^{n}d_k^2}{n(n^2-1)},\tag{1}$$

式中：

$$\bar{R}=\frac{1}{n}\sum_{k=1}^{n}R_k;\tag{2}$$

$$\bar{S} = \frac{1}{n}\sum_{k=1}^{n} S_k; \tag{3}$$

$$d_k = R_k - S_k。 \tag{4}$$

注: 1) r 的取值范围为 $[-1,1]$; 2) 当 $r>0$ 时,表明 X 和 Y 正相关,当 $r<0$ 时,表明 X 和 Y 负相关; 3) $|r|$ 越大,说明 X 和 Y 的相关程度越高; 4) 当 $r=1$ 时,则 X 和 Y 完全正相关,当 $r=-1$ 时,则 X 和 Y 完全负相关。

2.2 数据预处理

为了避免在数据处理过程中因某个或某些度量元数值差别过大而影响计算的精度,加快程序的收敛速度,需要对数据进行预处理。归一化就是将度量元属性数值规范到 $[0,1]$ 的过程,不影响原始数据属性对回归模型的预测。本文将单个维度下的度量元数据分别进行归一化,即

$$X' = \frac{X - \min(X)}{\max(X) - \min(X)}。 \tag{5}$$

另外,样本的排序对实验结果必然产生一定的影响,本文利用随机抽样的方法对数据顺序进行重新排列,避免因为训练集和测试集中个别实验数据异常而影响回归预测的整体效果,从而找出拟合度较高的回归模型。

2.3 缺陷密度预测模型构建

给定数据样本集合 $\{(x_1,y_1),\cdots,(x_i,y_i),\cdots,(x_l,y_l)\}$,其中,$x_i(i=1,2,\cdots,l)$ 为第 i 个软件项目的度量元向量,y_i 为第 i 个软件项目的缺陷密度,$x_i\in\mathbf{R}^n$,$y_i\in\mathbf{R}$。寻找 $\mathbf{R}^n$ 上的一个模型 $f(x)$ 来拟合这些点,以便用该模型推断任一输入 x 所对应的 y 值。其中,

$$f(x) = w\cdot x + b, \tag{6}$$

式中: w 为超平面的法向量; b 为常数。

回归预测的过程就是寻找一个最优超平面,使所有的样本点距离超平面的“总偏差”较小,因此采用一个正则化的误差函数进行度量,即

$$\frac{1}{2}\|w\|^2 + \frac{1}{2}\sum_{i=1}^{l}[f(x_i) - y_i]^2。 \tag{7}$$

式中: $f(x_i)$ 为第 i 个软件项目缺陷密度的预测值。

则原问题可转化为最优化问题:

$$\min_{w,\xi_i,\xi_i^*,b,\varepsilon}\ \frac{1}{2}\|w\|^2 + C\left[\nu\varepsilon + \frac{1}{l}\sum_{i=1}^{l}(\xi_i+\xi_i^*)\right], \tag{8}$$

$$\text{s.t.}\begin{cases}|(w\cdot x_i+b) - y_i| \leqslant \varepsilon+\xi_i,\\ \xi_i,\xi_i^* \geqslant 0,\varepsilon\geqslant 0。\end{cases} \tag{9}$$

式中: C 为惩罚因子; ε 为允许的范围误差; ν 为支持向量的个数; ξ_i、ξ_i^* 分别为训练样本对于超平面的上、下偏差。

引入拉格朗日乘子 α_i,得到对偶形式:

$$\max_{\alpha,\alpha^*}\sum_{i=1}^{l}[\alpha_i^*(y_i-\varepsilon) - \alpha_i(y_i+\varepsilon)] - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)K(x_i,x_j), \tag{10}$$

$$\text{s.t.}\begin{cases}\sum_{i=1}^{l}(\alpha_i-\alpha_i^*) = 0,\\ \sum_{i=1}^{l}(\alpha_i+\alpha_i^*) \leqslant C\cdot\nu,\\ 0\leqslant\alpha_i,\alpha_i^* \leqslant C/l,\end{cases} \tag{11}$$

式中: $K(x_i,x_j)$ 为核函数。

最终建立的 SVR 预测模型为

$$f(x) = \sum_{i=1}^{l}(\alpha_i-\alpha_i^*)K(x_i,x_j) + b。 \tag{12}$$

通过上述方法构建出软件缺陷密度预测模型,给定任一输入的度量元向量 x,则能够预测出软件项目所对应的缺陷密度 y。

2.4 核函数的选择

为了提高模型对缺陷密度预测的效果,首先需要对模型中的核函数进行选择。合适的核函数能够较好地将低维空间的非线性问题转化为高维空间的线性问题,在高维空间中寻找回归问题的最优超平面,解决了传统方法在转化过程中出现的计算量剧增的问题。

在缺陷密度的回归预测过程中,核函数的作用可表示为: 首先,把度量元所表示的向量 x 转化到高维空间的向量 x',并计算出高维空间中回归模型的 w' 和 b'; 然后,通过 $f(x') = w'\cdot x' + b'$ 进一步得到回归结果。核函数的目的就是建立数据从低维到高维转换的映射关系,当给定低维空间的 w 和 x 时,可直接输出确定的 $f(x')$。

模型中最常用的核函数主要有线性核函数、多项式核函数和 RBF 核函数[8-9]。其中,RBF 核函数是某种沿径向对称的标量函数,能够将数据映射到无穷维,同时能够逼近任意非线性函数,具有良好的泛化能力,收敛速度较快。因此,本文采用 RBF 核函数构建软件缺陷密度预测模型,其表达式为

$$K(x_i,x_j) = \exp\left[-\|x_i-x_j\|^2/(2\sigma)^2\right]。 \tag{13}$$

式中: x_j 为 RBF 核函数的中心; σ 为 RBF 核函数的宽度系数,决定着核函数的径向作用范围。

2.5 参数优化

虽然 SVR 模型中存在默认参数,但在实际问题

中并不能适用于所有问题的最优解，因此在研究具体问题时需要优化参数。参数优化包括惩罚因子 C 和 g 参数[10]的设置，其中：惩罚因子 C 在确定的特征空间中控制回归曲线的平滑度并折中经验风险；g 参数用来调节 RBF 核函数中的 σ，决定着 RBF 核函数的性能[11]。对于 C、g 这 2 个参数，国际上并没有给出公认统一的设置方法，需要根据具体研究对象和样本数据特点进行优化。

网格搜索法是一种遍历搜索的优化方法，其优化过程是依据步距将数值范围划分成网格，将所有数值分组并不断取任何可能的值。

参数优化的实质是针对不同的研究对象选择合适的 (C,g) 数据对。在缺陷密度的预测模型中设置 C、g 的数值范围，根据搜索步长将 C、g 分别划分为 n_1、n_2 组数值，然后将这些数值进行配对，共有 $n_1 \cdot n_2$ 组 (C,g) 数据对。网格搜索就是对 $n_1 \cdot n_2$ 组参数进行遍历搜索，通过验证选择出预测精度较高的 (C,g) 数据对作为最优参数。

验证过程采用十折交叉检验的方法，其主要过程是将所有的数据样本划分为 10 个子集，每个子集都将被其余数据训练后作为测试集。整个过程需要进行 10 次，直到每个子集都被当作一次测试集，最后将 10 次的平均交叉验证误差作为最终结果。网格搜索法通过设置搜索步长对参数进行全面搜索，与其他启发式或逼近的优化方法相比，搜索的范围更全面。由于每个 (C,g) 对是相互独立的，因此运算时可并行性高，能够节省一定的时间开销。

3　实验结果及分析

为验证基于 SVR 软件缺陷密度预测模型的效果，笔者基于 MATLAB2016a、VS2010 平台 C ++ 编译环境以及 LibSVM 工具包进行测试。实验采用我国特种车辆软件评测中心以及通用装备保障软件评测中心的 33 组实验数据进行测试。首先利用软件静态分析工具 LogiScope 对软件项目进行度量，采集 44 个软件缺陷度量元，通过 Spearman 相关系数分析法选择对缺陷密度影响较大的 5 种缺陷度量元进行回归预测，分别是耦合因子(ap_cof)、违反结构化编程数量(struc_pg)、使用某个类的类数量(cu_cdusers)、某个类使用的类数量(cu_cdused)和子类数量(in_noc)。然后与 Bagging、LinearRegression、Gaussian processes、IBK、MultilayerPerceptron 五种机器学习算法进行对比实验分析，5 种机器学习算法分别基于不同的原理构建预测模型，分析输入和输出之间的关系，算法具体描述如表 1 所示。

表 1　机器学习算法描述

算法	描述
Bagging	每次都通过采样来训练模型，泛化能力很强，对降低模型的方差有效
LinearRegression	根据一组变量 X 和其对应 Y 的样本来推测 X 和 Y 之间的线性关系
Gaussian processes	在特定条件下解决平方损失的优化问题
IBK	一般用于集合的预测分类过程
MultilayerPerceptron	前馈人工神经网络模型，其将输入的多个数据集映射到单一的输出数据集上

通过随机抽样的方法对 33 组数据进行重新排序后，分析预测结果，搜索预测结果较好的数据序列进行预测分析。将最优数据序列的前 22 组作为训练集，后 11 组数据作为测试集，验证网格搜索后预测模型的可行性和有效性。笔者采用均方根误差、平均绝对误差、相对平方根误差和相对绝对误差 4 个指标，分析预测的缺陷密度值和实际缺陷密度值之间的误差，各个指标的表达式如下：

均方根误差：

$$\varepsilon_1 = \left\{\frac{1}{n}\sum_{i=1}^{n}[f(x_i) - y_i]\right\}^{\frac{1}{2}}。\quad (14)$$

平均绝对误差：

$$\varepsilon_2 = \frac{1}{n}\sum_{i=1}^{n}|f(x_i) - y_i|。\quad (15)$$

相对平方误差：

$$\varepsilon_3 = \sum_{i=1}^{n}[f(x_i) - y_i]^2 / \sum_{i=1}^{n}(y_i - \bar{y})^2。\quad (16)$$

相对绝对误差：

$$\varepsilon_4 = \sum_{i=1}^{n}|f(x_i) - y_i| / \sum_{i=1}^{n}|y_i - \bar{y}|。\quad (17)$$

式中：$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$。

另外，通过相关系数衡量预测模型中度量元数据和缺陷密度的相关程度，相关系数越大，说明构建的预测模型效果越好。

支持向量回归共有 ε-SVR 和 ν-SVR 两种类型。下面通过实验分析采用 ε - SVR 和 ν - SVR 的不同预测效果，结果对比如表 2 所示。可知：支持向量回归中 ν-SVR 模型预测效果较好，这是因为 ν-SVR 模型在预测过程中能够自动将允许的范围误差 ε 最小化，而且能够控制支持向量的个数 ν，支持向量的个数直接决定预测的效果。

表2 不同SVR类型预测结果对比

SVR类型	ε_1	ε_2	ε_3	ε_4	相关系数
ε-SVR	0.155 9	0.148 3	0.950 5	1.284 9	0.463 4
ν-SVR	0.127 8	0.122 5	0.779 1	1.061 1	0.653 6

因此，笔者选取ν-SVR模型进行实验，并将参数优化后的SVR预测模型与5种机器学习算法的预测模型进行对比分析，其缺陷密度预测结果对比如表3所示。

表3 不同预测模型缺陷密度预测结果对比

模型	ε_1	ε_2	ε_3	ε_4	相关系数
本文SVR	0.127 8	0.122 5	0.779 1	1.061 1	0.653 6
Bagging	1.442 2	1.258 3	0.926 6	0.848 0	0.564 6
LinearRerression	3.365 2	2.388 4	2.162 0	1.609 5	0.422 4
Gaussian processes	2.201 4	1.823 7	1.414 3	1.229 0	0.463 0
IBK	1.575 1	1.170 3	1.011 9	0.788 6	0.590 9
MultilayerPerceptron	2.478 2	2.033 7	1.592 2	1.370 5	0.438 7

由表3可见：与基于5种机器学习算法构建的预测模型相比，本文的SVR预测模型得到的缺陷密度预测误差较小，度量元数据和缺陷密度的相关系数较大，表明该模型的预测效果较好。

通过网格搜索法对SVR预测模型参数寻优后，得到C的最优值为8，g的最优值为0.5，结果如图2所示。

测试集模型的预测结果如图3所示，可以看出：

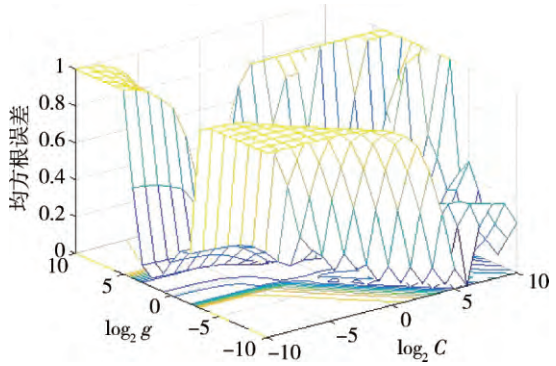

图2 SVR参数寻优结果

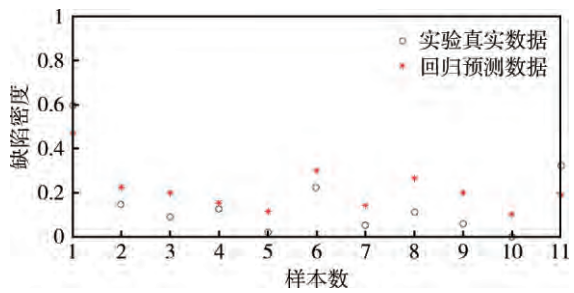

图3 测试集模型的预测结果

缺陷密度回归预测值基本能够拟合实际缺陷密度值，误差在合理范围之内。

4 结论

笔者通过Spearman相关系数提取对缺陷密度影响较大的缺陷度量元数据，利用SVR构建软件缺陷密度预测模型，并采用网格搜索法对模型中参数(C,g)进行优化，一定程度上提高了回归预测的效果，最后与5种机器学习算法进行了对比实验，结果表明：所建立的SVR预测模型是有效的，能较好地预测软件缺陷。

参考文献：

[1] 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(1)：1-25.

[2] AGARWAL S，TOMAR D，Prediction of software defects using twin support vector machine[C]//International conference on information systems and computer networks. Mattura：IEEE，2014：128-132.

[3] SUFFIAN M D M，IBRAHIM S. A prediction model for system testing defects using regression analysis[J]. International journal of soft computing & software engineering，2012，2(7)：55-68.

[4] OKUTAN A，YILDIZ O T. Software defect prediction using Bayesian networks[J]. Empirical software engineering，2014，19(1)：154-181.

[5] 王李进，吴保国，郑德祥. 基于人工神经网络的软件质量评价[J]. 计算机应用与软件，2008，25(12)：133-134，150.

[6] 吴晓萍，赵学靖，乔辉，等. 基于LASSO-SVM的软件缺陷预测模型研究[J]. 计算机应用研究，2013，30(9)：2748-2751，2754.

[7] 姜慧研，宗茂，刘相莹. 基于ACO-SVM的软件缺陷预测模型的研究[J]. 计算机学报，2011，34(6)：1148-1154.

[8] 王男帅，薛静锋，胡昌振，等. 基于遗传优化支持向量机的软件缺陷预测模型[J]. 中国科技论文，2015(2)：159-163.

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

[10] RYU D，CHOI O，BAIK J. Value-cognitive boosting with a support vector machine for cross-project defect prediction[J]. Empirical software engineering，2016，21(1)：43-71.

[11] SHAN C，ZHU H J，HU C Z，et al. Software defect prediction model based on improved LLE-SVM[C]//International Conference on Computer Science and Network Technology. Harbin：IEEE，2015：530-535.

（责任编辑：尚彩娟）