研究与开发

基于最大信息系数的软件缺陷数目预测特征选择方法

刘国庆，王兴起，魏丹，方景龙，邵艳利
（杭州电子科技大学计算机学院，浙江 杭州 310018）

摘　要：针对传统特征选择方法仅考虑变量间的线性关系而忽略非线性相关性，导致软件缺陷数目预测模型的性能较低的问题，提出了一种基于最大信息系数的特征选择方法。该方法考虑特征与特征以及特征与缺陷数目间的线性及非线性关系，将特征的冗余性分析和相关性分析分离为两个阶段。在冗余特征分析阶段，基于特征间的相关度，采用凝聚层次聚类算法将冗余特征分到同一簇中；在相关性分析阶段，依据特征与软件缺陷数目之间的相关度，对每个特征簇中的特征进行排序，然后从簇中选择排名靠前的特征组成特征子集。实验结果表明，该方法能够选择有效的特征子集，提高软件缺陷数目预测模型的预测性能。

关键词：软件缺陷数目预测；特征选择；最大信息系数
中图分类号：TP311
文献标识码：A
doi: 10.11959/j.issn.1000−0801.2021025

Feature selection method for software defect number prediction based on maximum information coefficient

LIU Guoqing, WANG Xingqi, WEI Dan, FANG Jinglong, SHAO Yanli
School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China

Abstract: The traditional feature selection method only considers the linear correlation between variables and ignores the nonlinear correlation, so it is difficult to select effective feature subsets to build the effective model to predict the number of faults in software modules. Considering the linear and nonlinear relationship, a feature selection method based on maximum information coefficient (MIC) was proposed. The proposed method separated the redundancy analysis and correlation analysis into two phases. In the previous phase, the cluster algorithm, which was based on the correlation between features, was used to divide the redundant features into the same cluster. In the later phase, the features in each cluster were sorted in descending order according to the correlation between features and the number of software defects, and then the top features were selected to form the feature subset. The experimental results show that the proposed method can improve the prediction performance of software defect number prediction model by effectively removing redundant and irrelevant features.

Key words: software defect number prediction, feature selection, maximum information coefficient

收稿日期：2020−06−03；**修回日期：**2021−01−18
通信作者：方景龙，fjl@hdu.edu.cn
基金项目：浙江省自然科学基金资助项目（No.LY20F020015，No.LY21F020015）；国家自然科学基金资助项目（No.61702517，No.61972121，No.61702146）；国防基础科研计划资助项目（No.JCKY2019415C001）
Foundation Items: The Natural Science Foundation of Zhejiang Provincial of China (No.LY20F020015, No.LY21F020015), The National Natural Science Foundation of China (No.61702517, No.61972121, No.61702146), Defense Industrial Technology Development Program (No.JCKY2019415C001)

1 引言

随着信息时代的到来，计算机软件数量、种类快速增长，软件质量问题被越来越多的人关注。在软件开发的过程中，对软件需求的错误理解或人员的经验不足都可能引入缺陷。软件缺陷是软件质量的对立面，威胁着软件质量[1]。从软件开发的角度来看，处理软件缺陷是一项至关重要的任务。缺陷的存在不仅降低了软件的质量，而且还增加了软件的开发、测试和维护成本。因此，在软件开发的早期阶段预测出软件缺陷模块，可以指导软件测试人员关注缺陷模块，有助于提高软件质量。

目前在软件缺陷预测领域已经有大量研究工作，绝大多数致力于识别软件模块中有缺陷或无缺陷，含有不同缺陷数量的软件模块在软件测试中被同等对待。然而，对于有缺陷的模块，有些软件模块含有的缺陷较多，有些模块含有的缺陷数较少，与含有较少缺陷的模块相比，缺陷数量相对较多的软件模块则需要付出更多的资源去测试和修改。因此，仅根据有缺陷和无缺陷分配质量保证资源可能会导致资源利用效率低下，而预测软件模块的缺陷数目可以给软件测试提供更多的参考，更有利于分配测试资源。

目前，已有工作提出并且评估了基于回归模型的软件缺陷数目预测的方法，这类方法可以预测软件模块中有多少缺陷。但大部分都是基于模型的角度，利用从软件历史仓库挖掘的数据集，通过对比不同的回归模型，得到一个性能较好的回归模型。在构建缺陷数目预测模型时，数据集中的度量元含有冗余特征或无关特征，这些特征的存在会严重影响预测模型的性能。因此，设计有效的特征选择方法来移除这类特征具有重要意义。

特征相关性度量标准的选取对于冗余特征和无关特征的衡量非常重要。目前软件缺陷预测中常见的特征度量[2]大多仅能度量特征间的线性关系，难以刻画特征间的非线性关系。在软件缺陷数据中特征之间既存在线性相关关系，也存在非线性相关关系。对于软件度量特征来说，非线性的相关度度量方法比线性方法更为实际有效[3-4]。最大信息系数（maximum information coefficient，MIC）能够广泛地度量变量间的依赖关系，包括线性关系、非线性关系以及非函数依赖关系。

本文提出了一种基于 MIC 的两阶段特征选择方法（two-stage feature selection method based on maximum information coefficient，MIC-TSFS），该方法基于 MIC 度量特征与特征之间、特征与软件缺陷数目之间的关联性，衡量特征之间线性和非线性相关性，从而建立将特征冗余性分析和相关性性分析相分离的两阶段特征选择框架。为验证 MIC-TSFS 方法的有效性，在 PROMISE 真实软件项目上进行了实证性的研究，实验结果表明，MIC-TSFS 能够移除冗余特征和无关特征，得到优化的特征子集，构建更有效的软件缺陷数目预测模型。

2 相关工作

软件缺陷数目预测从软件历史仓库中得到历史缺陷数据，然后构建缺陷数目预测模型，并用模型来预测出被测项目中软件模块的缺陷数目，从而为软件测试中如何分配测试资源提供指导意见。软件缺陷数目预测是智能化软件工程领域中的一个研究热点。

目前，国内外的研究者提出了多种有效的软件缺陷数目预测模型。Graves 等[3]建立线性回归（linear regression，LR）模型并应用于大型的电信系统项目。Ostrand 等[4]使用负二项回归（negative binomial regression，NBR）技术对故障数和故障密度预测进行了研究。Chen 等[5]探究了一些典型的回归模型在软件缺陷个数预测上的可行性，实验结果表明，决策树回归取得了较好预测结果。Rathore 等[6]提出了一种基于遗传编程的软件缺陷预测方法，实验结果表明，利用该方法对缺陷数目进行预测能够取得较好结果。Rathore 等[7]探究

了决策树回归模型在版本内缺陷数目预测和跨版本缺陷数目预测两种不同的研究场景下的预测能力，结果表明该方法在版本内进行预测时能够取得较好的结果。Rathore[8]比较了 6 种用于软件缺陷数目预测的方法（遗传编程（genetic programming）、多层感知（multi-layer perceptron）、线性回归（linear regression）、决策树回归（decision tree regression）、零膨胀泊松回归 （zero-inflated Poisson regression）和负二项回归 （negative binomial regression））对于软件缺陷数目的预测能力，实验结果表明决策树回归、遗传编程、多层感知和线性回归通常会有较好的预测性能，而零膨胀泊松回归和负二项回归表现通常不佳。Chen 等[9]探究了有监督方法与无监督方法在缺陷数目预测方面的能力，将 9 种结合 SMOTEND 的有监督方法与基于 LOC 度量的无监督方法在版本内预测、跨版本预测、跨项目预测中分别进行了比较。

除上述基础模型外，也有些研究工作针对缺陷数目预测提出了一些集成方法。Rathore 等[10]研究了集成学习方法在软件缺陷数预测方面的性能，先评价 6 种基础模型，然后从中选择 3 种性能较优的模型进行集成。他们[11]还提出了两种线性规则、两种非线性规则，用于集成基础模型的输出。并且做了实证研究，评估了在版本内预测以及版本间预测两种情景模式下，各个集成规则的预测性能。Yu 等[12]探索了采样（SMOTEND 和 RUSND）和集成学习（AdaBoost.R2）方法对软件缺陷数目的预测能力，然后提出了 SMOTENDBoost 和 RUSNDBoost 这两种混合方法，实验结果表明融合过采样与集成学习可以有效提高对于缺陷数量的预测性能。

以上研究都是从模型的构建的角度出发，致力于找到一个性能较好的预测模型。然而，在构建软件缺陷数目预测模型时，在软件缺陷的大量度量元中不可避免地会产生冗余特征和无关特征。冗余特征大量或完全重复了其他单个或多个特征中含有的信息，而无关特征则不能对采用的数据挖掘算法提供任何帮助。若在构建模型之前去除冗余特征和无关特征，则能在一定程度上提升模型的性能。

特征选择通过挖掘特征间以及特征与缺陷数目间的内在联系，保留最有利于预测的有效特征，是除去冗余特征和无关特征的一种有效的方法。除去冗余特征和无关特征时需要衡量变量之间的相关性，信息增益[13]、ReliefF 等是衡量相关性的常见方法。目前已有的特征选择算法可以分为两大类：基于子集搜索的特征选择算法和基于排序的特征选择算法。基于子集搜索的特征选择算法考虑特征之间的冗余性和特征与类别之间的相关性，但这类方法需要搜索的特征子集空间太大，计算开销过大；基于排序的特征选择算法根据特征与类别之间的相关性从高到低排序，排名越靠前，与类别属性的相关性越高，表明该特征区分不同类别的能力越强，然后从中选出排名靠前的特征来构建预测模型，其运行效率比较高，但选出的特征子集内通常含有冗余特征。

目前已有研究人员将特征选择方法应用到软件缺陷数目预测问题中。李叶飞等[14]提出了一种针对软件缺陷数目预测的特征选择方法——FSDNP，该方法采用欧氏距离度量特征间相关性，使用密度峰聚类对特征进行聚类，结合采用 Pearson 相关系数计算特征与缺陷数目之间的相关性时表现性能较好。Yu 等[15]提出软件缺陷预测下的特征选择方法——FSCR，该方法首先基于特征间的 Pearson 相关系数进行谱聚类，然后根据 ReliefF 算法挑选与缺陷数目相关的特征。马子逸等[16]提出了一种面向软件缺陷个数预测的混合式特征选择方法——HFSNFP，该方法先使用 ReliefF 算法计算特征与缺陷个数之间的相关性，移除无关特征，随后根据特征与特征之间的 Pearson 相关系数对特征集进行谱聚类，将冗余度高的特征聚类到同一个簇中，最后基于包裹式特征选择的思想依次

从每个簇中将最相关的特征加入特征子集中。这类针对缺陷数目预测而提出的特征选择方法在衡量特征与特征之间的相关性时通常采用 Pearson 相关系数或距离度量，仅表示了特征间的线性关系，忽略了特征之间的非线性相关性，不利于特征间的冗余性分析；此外，这类方法中采用的聚类方法需要事先设定参数，这些设定具有很大的主观性和经验性，设定不当会影响预测模型的性能。

基于以上分析，为了有效地识别并移除特征集中的冗余特征和无关特征，本文提出了一种基于最大信息系数的特征选择方法，该方法包括冗余性分析阶段和相关性分析阶段。在冗余性分析阶段，根据特征与特征间的关联度，采用自底向上的层次聚类的算法对特征进行聚类得到特征簇，降低所选特征子集的冗余率；在相关性分析阶段，根据特征与缺陷数目之间的关联度将每个特征簇中的特征进行排序，然后从每个特征簇中选择排名靠前的特征组成特征子集，过滤掉无关特征。

3 基于最大信息系数的特征选择方法

3.1 最大信息系数

最大信息系数可以用于衡量两变量的相关程度，其主要思想是：如果两个变量之间存在一定的关联，在这两个变量的散点图上进行某种网格划分之后，根据这两个变量在网格中的近似概率密度分布，计算这两个变量的互信息，该值经过正则化后可用于度量这两个变量之间的相关性。最大信息系数是互信息的推广，不仅可以度量变量间的线性关系，而且可以度量变量间的非线性关系和非函数依赖关系。

给定变量 $\boldsymbol{X}=\{x_i, i=1,2,3,\cdots,n\}$，$\boldsymbol{Y}=\{y_i, i=1,2,3,\cdots,n\}$，$n$ 为样本数量，其互信息定义如下：

$$\mathrm{MI}(X,Y)=\sum_{x,y}p(x,y)\log\frac{p(x,y)}{p(x)p(y)} \qquad (1)$$

假设有限集合 D 由 X 和 Y 组成，定义划分 G 将变量 X 的值域分成 x 段，将 Y 的值域分成 y 段，得到一个 $x\times y$ 的网格，在得到的每一种网格划分内部计算互信息 $\mathrm{MI}(X,Y)$，取不同划分的最大互信息作为划分 G 的互信息值，定义划分 G 下 D 的最大互信息公式为：

$$\mathrm{MI}^*(D,x,y)=\max \mathrm{MI}(D\,|\,G) \qquad (2)$$

其中，$D|G$ 表示数据 D 使用 G 进行划分。

对于不同的划分都会得到一个 MI 值，将所有 MI 值组成特征矩阵，并对该特征矩阵进行规范化并定义为 $M(D)_{x,y}$，其计算式如下：

$$M(D)_{x,y}=\frac{\mathrm{MI}^*(D,x,y)}{\log\min\{x,y\}} \qquad (3)$$

$M(D)_{x,y}$ 为 D 在不同网格划分下的规范化后的特征矩阵。最大信息系数 MIC 的计算式如下：

$$\mathrm{MIC}(D)=\max_{xy<B(n)}\{M(D)x,y\} \qquad (4)$$

其中，$B(n)$ 为网络划分 $x\times y$ 的上限值，一般地，$1\leqslant B(n)\leqslant O(n^{1-\varepsilon}), 0<\varepsilon<1$，参考文献[17]中给出 $B(n)=n^{0.6}$ 时效果较好。

MIC 通过对连续型变量实施不等间隔的离散化寻优来挖掘非线性关联，并进一步通过标准化使得 $\mathrm{MIC}(X,Y)\in[0,1]$。如果 X、Y 相互独立，MIC(X,Y)=0，MIC 值越大，两个特征越相关；MIC 具有对称性，即 MIC(X,Y)=MIC(Y,X)；当 $Y=f(X)$ 时，MIC(X,Y)=1，即当两个变量之间存在函数关系时，这两个变量之间的 MIC 值为 1；若对变量 X 做任何单调函数 $g(X)$ 变换，MIC 不变，即 MIC(X,Y)=MIC(g(X),Y)。

在软件缺陷预测领域中，为了充分描述软件特性，度量元特征集中存在冗余特征和无关特征。一些特征由同一组程序基本属性计算得到，这些特征之间有很大概率存在冗余。缺陷数据特征之间不仅存在线性关系（如变量个数与代码行数），也存在非线性关系（如类个数与类继承树深度）。一般的相关系数（如 Pearson 相关系数）因仅表示特征间的线性关系，不能充分表征特征间的相关性；而 MIC 具有较好的广泛性和均匀性，能够检测出变

量间的线性和非线性关系，进而有效表达特征之间的相关性。本文把 MIC 作为特征与特征间相关性度量判据，从而通过移除冗余特征初步筛选特征集。

无关特征是与预测目标不相关的特征，特征子集的优劣可以通过特征与缺陷数目的相关性来度量。一般认为，好的特征子集中的特征与缺陷数目相关度较高。在传统的特征选择方法中，若特征与缺陷数目间存在非线性函数依赖关系，很有可能被排除在模型外；而 MIC 受噪声影响较小，可以敏感地检测出变量之间的各种函数以及非函数关系。另外，一般的相关系数对变量变换很敏感，然而在构建预测模型时变换并不明确，相关系数容易受到影响；而无论做何种单调变换，变量间的 MIC 值不变。因此，MIC 稳健性更好，能够有效地度量特征与缺陷数目之间的相依关系。

基于上述分析，用 MIC 来选择特征适用于有复杂的软件缺陷的数据。因此，本文采用 MIC 衡量特征与特征间的相关性、特征与缺陷数目间的相关性，以此来移除冗余特征、初步筛选特征集，随后剔除一些无关特征，进而提高模型的预测性能。

3.2　特征关联性度量

给定一个 n 条样本的数据集 $F\{f_1,f_2,f_3,\cdots,f_N,d\}$，其中 $f_i(i=1,2,3,\cdots,N)$ 为特征集中第 i 个特征，N 为特征总数，d 为软件缺陷数目。对任意特征 f_{i} 和 f_{j}，它们之间的相关度定义为 $\mathrm{FF}_{f_if_j}$：

$$\mathrm{FF}_{f_if_j}=\mathrm{MIC}(f_i,f_j) \tag{5}$$

在软件预测数据集中，特征间的冗余性越大，则它们相关度值越大。$\mathrm{FF}_{f_if_j}$ 值越大，说明 f_i 和 f_j 之间的可替代性越强，即冗余性越强，聚类时趋向于聚在同一个类簇中。$\mathrm{FF}_{f_if_j}$ 值越小，说明 f_i 和 f_j 之间越相互独立，聚类时趋向与聚在不同的簇类中。

特征 f_i 与软件缺陷数目 d 之间的相关度定义为 FD_i：

$$\mathrm{FD}_i=\mathrm{MIC}(f_i,d) \tag{6}$$

FD_i 值越大表明特征 f_i 与软件缺陷数目 d 间的相关性越强，则 f_i 为强相关特征，越倾向被保留；FD_i 值越小，表明特征 f_i 与软件缺陷数目 d 的相关性越弱，则 f_i 为弱相关特征，越倾向被删除。

3.3　MIC-TSFS 方法

MIC-TSFS 是一种将特征冗余性分析和相关性分析分离的两个阶段特征选择方法，其总体框架如图 1 所示。

第一阶段是冗余性分析阶段，根据特征与特征之间的相关度，采用层次聚类将原始数据集中的特征进行聚类，使得相关性较高的特征被划分到同一簇中，而相关性较低的特征被划分到不同簇中，这样同一个簇里的特征冗余性较高，不同簇中的特征冗余性较低。将这一阶段处理得到的特征簇作为第二阶段的输入。第二阶段是相关性分析阶段，在每个特征簇中计算特征与软件缺陷数目的相关度，根据该值对特征簇内的特征进行排序，在每个簇中选取排名靠前的特征组成新的特征子集。

3.3.1　冗余性分析阶段

为了降低特征间的冗余性，根据特征与特征之间的相关度，采用“自底向上”的凝聚层次聚类将数据集中的特征集合进行聚类，将特征分为若干个特征簇，同一簇中的特征相关度高，不同簇的特征相关度低。相对于最常用的密度峰聚类和谱聚类算法，层次聚类算法具有对聚类参数依赖小、适用于任意形状数据集聚类的优势。因此，MIC-TSFS 在

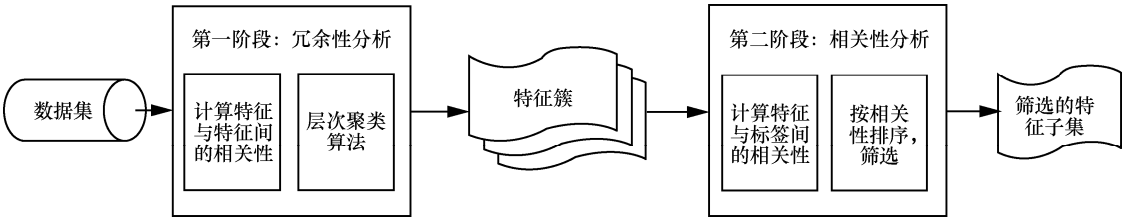

图 1　MIC-TSFS 框架

冗余性分析阶段选取了更适用于构建鲁棒特征选择算法的凝聚层次聚类算法。凝聚层次聚类算法过程是：初始时将每个样本看作一个类簇，然后依据某种准则合并这些初始的类簇，直到达到某种条件或者达到设定的分类数目，具体如算法 1 所示。

算法 1 凝聚层次聚类算法

将样本集中的所有的样本点都当作一个独立的类簇

repeat

计算两两簇之间的相似性，找到最相似的两个簇

将最相似的两个簇合并为一个簇

util

达到聚类的数目或者达到设定的条件

在算法 1 中，有多种方法可以计算簇间相似度，包括最小相似度、最大相似度和均值相似度，由于最小和最大相似度量代表了簇间相似度量的两个极端，它们趋向对噪声数据过分敏感，因此，MIC-TSFS 采用均值来度量簇间相似度。给定两个特征簇 C_i 和 C_j，簇 C_i 中包含 p 个特征，表示为 $\{f_1^{c_i}, f_2^{c_i},\cdots,f_p^{c_i}\}$，簇 C_j 中包含 q 个特征 $\{f_1^{c_j},f_2^{c_j},\cdots,f_q^{c_j}\}$，采用如下计算式计算它们之间的相关度：

$$S_{C_iC_j}=\frac{1}{p\times q}\sum_{i'=1}^{p}\sum_{j'=1}^{q}FF_{f_{i'}^{C_i}f_{j'}^{C_j}} \tag{7}$$

若不提前中止簇合并，算法 1 会将所有的特征合并到同一个簇中。在算法 1 中，可设置达到聚类的数目或者达到设定的条件，使合并的过程提前停止。为了避免参数依赖问题，MIC-TSFS 通过自适应调参方式设置合并中止条件：

$$r=\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}FF_{f_if_j} \tag{8}$$

其中，N 为特征的个数。当迭代过程中拟合并的两个簇之间的相似度小于 r 时，簇与簇之间的相似度较小，不再进行簇间的合并，从而结束迭代过程。

3.3.2 相关性分析阶段

在相关性分析阶段，为了过滤掉与软件缺陷数目无关的特征，选择对软件缺陷数目预测较为有用的特征。依据特征和软件缺陷数目之间的相关度，对每个特征簇中的特征进行排序，然后依据簇的大小从每个特征簇中挑选排名靠前的特征。具体步骤如下：

（1）对于每个特征簇，逐个计算簇内特征 f_i 与软件缺陷数目 d 之间的相关度 FD_i；

（2）从每个特征簇中选取前 M 个特征。对于 M 的取值，参考文献[18]认为将软件缺陷数据集中特征数目减少为 $\lceil \mathrm{lb}N \rceil$ 较为合适。考虑到簇中特征数目可能存在为 1 的情况，此时 $\lceil \mathrm{lb}1 \rceil=0$，会导致此特征将无法被有效处理，因此，MIC-TSFS 设置 M 的取值为 $\lfloor \mathrm{lb}p \rfloor+1$，其中 p 为每个簇的特征个数。

3.4 算法描述

MIC-TSFS 特征选择方法的实现过程如算法 2 所示。

算法 2 MIC-TSFS 算法

输入 $F\{f_1,f_2,f_3,\cdots,f_N,d\}$：原始数据集

输出 $F_{\mathrm{sub}}\{f_{i1},f_{i2},f_{i3},\cdots,f_{im}\}$：最终选择出来的特征子集

冗余性分析阶段

（1）初始化特征簇，对于含有 N 个特征的特征集 $\{f_1,f_2,f_3,\cdots,f_N\}$，每个特征作为一个簇，共有 N 个簇。

（2）根据式（8）计算阈值 r，作为聚类结束的判断条件。

（3）根据式（7）计算簇与簇之间的相似度，选取簇间相关度最大两个簇 C_i 和 C_j。

（4）while $S_{C_iC_j}>r$

合并这两个簇；

根据式（7）重新计算簇与簇之间的相似度，选择相关度最大两个簇。

（5）聚类结束，聚类结果为 $\{S_1,S_2,\cdots,S_{N'}\}$，共有 N' 个簇。

相关性分析阶段

（6）初始化$F_{\text{sub}}\{\}$为空集。

（7）for i in 1： N' do。

（8）根据式（6），计算簇$C_i\{f_1^{c_i},f_2^{c_i},\cdots,f_p^{c_i}\}$中的特征$f_i^{c_i}(i=1,2,3,\cdots,p)$与软件缺陷数目 d 的相关性，并根据相关性的值对簇C_i的特征从大到小排序，排序后结果表示为：$C_i'\{f_{i_1},f_{i_2},\cdots,f_{i_p}\}$。

（9）从C_i'中选取前 M 个特征并入 F_{sub}

（10）end for

（11）return F_{sub}

步骤（1）～步骤（5）为使用特征间相关度与凝聚层次聚类算法对冗余特征的处理过程，经过聚类之后，将冗余特征分到同一个特征簇中，即每个特征簇中包含大量的冗余特征，将在第二阶段进行删除。步骤（6）～步骤（8），在每个特征簇中计算特征与软件缺陷数目间的相关度，然后按照此相关度的值对特征进行降序排序。步骤（9）选取每个特征簇中排名靠前的特征组成特征子集。经过 MIC-TSFS 处理之后，原始数据集中的冗余特征和不相关特征将被过滤，从而得到高质量特征子集，该特征子集可以作为回归模型的输入构建软件缺陷数目预测模型。

4　实证研究

4.1　评测对象

从软件缺陷常用数据 PROMISE 项目集[10]中选取 4 个项目共 14 个常用公开数据集进行实证研究，这 4 个项目均是 Apache 的实际项目。其中 Ant 是 Apache 中一个由 Java 语言进行编写的子项目；Camel 是 Apache 的一个开源项目，提供基于规则的路由引擎；Xerces 是一项用于 XML 文档解析的开源项目；ivy 是 Ant 的子项目，主要用来解决 Ant 的 jar 包的版本管理。对各个版本的项目源程序按照参考文献[19]的度量方式进行度量，从而得到项目中每个模块的特征。每个项目都含有很多类级别的特征，例如特征 dit 表示一个类的继承数的深度；特征 noc 表示一个类的直接子类的个数；特征 loc 表示类的二进制码的行数等。这些特征都是基于代码复杂度和面向对象特性进行设计的。数据集特征见表 1，Release 表示软件项目不同的发行版本，#Instance 表示实例的数量，#Defects 表示实例中缺陷的总数，%Defects 表示有缺陷的实例占所有实例的百分比，Max 表示所有实例中最大的缺陷个数。这些项目在之前的研究工作中被广泛使用，并可从 PROMISE 中免费获取。

表 1　数据集特征

project	Release	#Instance	#Defects	%Defects	Max
ant	ant-1.3	125	33	16%	3
	ant-1.4	178	47	22.50%	3
	ant-1.5	293	35	10.90%	2
	ant-1.6	351	184	26.20%	10
	ant-1.7	754	338	22.30%	10
camel	camel-1.0	339	14	3.80%	2
	camel-1.2	608	522	35.50%	28
	camel-1.4	872	335	16.60%	17
	camel-1.6	965	500	19.50%	28
xerces	xerces-1.2	453	193	15.20%	30
	xerces-1.3	588	1 596	74.30%	62
ivy	ivy-1.1	111	233	56.80%	36
	ivy-1.4	241	18	6.70%	3
	ivy-2.0	352	56	11.40%	3

4.2　评测指标

为了评估 MIC-TSFS 在软件缺陷数目预测中的性能，使用平均绝对误差（average absolute error，AAE）、平均相对误差（average relative error，ARE）、G-mean 评价指标。

AAE 是一组预测数据中所有误差的平均值，它表明了预测结果和真实结果之间的差距。其定义如下：

$$\text{AAE}=(1/n)\sum_{i=1}^{n}\left|\overline{Y_i}-Y_i\right| \tag{9}$$

其中，ARE 表示当整体目标项目被评估时相对误差的大小，其定义如下：

$$\mathrm{ARE}=(1/n)\sum_{i=1}^{n}\left|\overline{Y_i}-Y_i\right|/(Y_i+1) \qquad (10)$$

其中，$\overline{Y_i}$ 表示测试集中的第 i 个样本的预测缺陷数目，Y_i 表示测试集中的第 i 个样本真实含有的缺陷数目，n 表示测试集中的样本数目。

为了进一步比较各种方法之间的性能，本文使用“Win/Draw/Loss”分析：“Win”表示“方法 1”好于“方法 2”的次数；“Draw”表示“方法 1”相似与“方法 2”的次数；“Loss”表示“方法 1”差于“方法 2”的次数。

在软件缺陷数目预测中，平均绝对误差与平均相对误差反映了预测值偏离真实值的程度。为了衡量预测的准确性，引入 G-mean 作为评价指标，其定义如下：

$$\text{G-mean}=\sqrt{\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}\times\frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}}} \qquad (11)$$

TP/(TP+FN)描述模型能正确搜索缺陷数目大于 0 的样本的能力；TN/(TN+FP)描述模型能正确搜索缺陷数目等于 0 的样本的能力；G-mean 是这两种能力的综合评价指标，其值越大，表示模型预测性能越好。其中，TP 是指被正确预测的有缺陷的样本数，FP 是指被错误预测为有缺陷的无缺陷的样本数，FN 是指被错误预测为无缺陷的有缺陷的样本数，TN 是指被正确预测的无缺陷的样本数。本文预测软件缺陷数目，即缺陷数目大于 0 的实例被认为有缺陷，缺陷数目等于 0 即无缺陷。

4.3 回归模型

本文采用了 3 种回归模型来构建软件缺陷数目预测模型：贝叶斯岭回归（Bayesian ridge regression）、线性回归（linear regression）和决策树回归（decision tree regression）。这 3 种回归模型是在软件缺陷数目预测中常用而且性能较优的模型[17]。

- 贝叶斯岭回归：该方法类似于经典的岭回归方法。这种方法的超参数是通过先验概率引入的，然后基于概率模型的最大化边缘对数似然来评估。
- 线性回归：它通常被用于求解一个或多个自变量和一个因变量之间的线性关系的最小二乘函数。
- 决策树回归：该方法学习简单的决策规则来近似拟合给定的训练数据，然后预测目标变量。

在实验中，采用 10 折交叉验证，即将原始数据集等分为 10 份，使用其中的 9 份作为训练集，剩余的 1 份作为测试集，反复 10 次。在此过程中，每 1 份数据都保证恰好有一次作为测试集，最后结果采用 10 次交叉实验结果的平均值。

4.4 结果分析

在 PROMISE 数据集上对所提出的方法进行了实证性研究，对比了 ReliefF 算法、信息增益、卡方检验 3 种传统的特征选择方法，以及 HFSNFP[18]、FSCR[17]、FSDNP[16]3 种用于软件缺陷数目预测的特征选择方法。对于传统的特征选择方法，本文选取的特征数为 $\lfloor 1\mathrm{b}N \rfloor+1$，其中 N 为特征的总数量。对于 HFSNFP、FSCR 和 FSDNP 方法，本文分别根据参考文献[16-18]中的设置来选择特征。在贝叶斯岭回归模型中，全特征（full）、ReliefF 算法、信息增益（info）、卡方检验（chi2）、HFSNFP、FSCR、FSDNP 等方法与 MIC-TSFS 方法的 AAE 见表 2。

在贝叶斯岭回归模型中全特征（full）、ReliefF 算法、信息增益（info）、卡方检验（chi2）、HFSNFP、FSCR、FSDNP 等方法与 MIC-TSFS 方法的 ARE 的对比结果见表 3。在表 2 中，第 1 列表示项目的名称，剩余列表示各种方法的 AAE 平均值。最后一行表示 MIC-TSFS 方法与其他方法之间的“Win/Draw/Loss”比较结果。例如 MIC-TSFS 与全特征的结果为“10/1/3”，即在 10 个数据集上 MIC-TSFS 比全特征的性能较好，在 1 个数据集上 MIC-TSFS 与全特征性能相似，在 3 个数据集上 MIC-TSFS 比全特征的性能弱。贝叶斯岭回归模型下，MIC-TSFS 与传统的特征选择方法 ReliefF、

表 2　贝叶斯岭回归模型的 AAE

	full	ReliefF	info	chi2	HFSNFP	FSCR	FSDNP	MIC-TSFS
ant-1.3	0.301 3	0.287 2	0.278 8	0.301 9	0.278 8	0.277 6	0.302 6	0.270 5
ant-1.4	0.281 4	0.281 7	0.275 8	0.287 3	0.276 1	0.270 3	0.264 7	0.270 3
ant-1.5	0.115 7	0.123 1	0.119 8	0.119 3	0.112 9	0.119 7	0.133 6	0.116 2
ant-1.6	0.481 7	0.447 5	0.641 5	0.461 8	0.533 2	0.461 7	0.464 2	0.456 0
ant-1.7	0.370 5	0.598 5	0.551 8	0.398 7	0.463 1	0.459 1	0.422 8	0.398 7
camel-1.0	0.044 2	0.044 2	0.044 2	0.044 2	0.044 2	0.044 2	0.044 2	0.044 2
camel-1.2	1.038 9	1.060 6	1.163 9	1.075 0	1.091 4	1.040 4	1.048 8	1.029 0
camel-1.4	0.477 1	0.506 9	0.522 9	0.463 3	0.482 8	0.494 3	0.478 3	0.466 7
camel-1.6	0.704 4	0.703 2	0.696 0	0.713 7	0.695 9	0.716 5	0.700 3	0.688 8
xerces-1.2	0.281 8	0.261 4	0.263 6	0.281 8	0.270 5	0.268 2	0.275 0	0.277 3
xerces-1.3	0.545 1	0.544 5	0.575 6	0.566 8	0.539 9	0.535 8	0.597 3	0.538 3
ivy-1.1	1.312 1	1.376 5	1.896 2	1.312 9	1.372 0	1.840 2	1.312 9	1.248 5
ivy-1.4	0.082 7	0.070 2	0.082 7	0.082 7	0.078 5	0.082 7	0.082 7	0.091 0
ivy-2.0	0.173 2	0.176 2	0.158 8	0.170 5	0.178 8	0.164 7	0.167 5	0.161 8
mean	0.443 6	0.463 0	0.519 4	0.448 6	0.458 4	0.483 2	0.449 6	**0.432 7**
Win/Draw/Loss	10/1/3	10/1/3	10/1/3	10/2/2	10/1/3	9/2/3	9/1/4	

表 3　贝叶斯岭归模型的 ARE

	full	ReliefF	info	chi2	HFSNFP	FSCR	FSDNP	MIC-TSFS
ant-1.3	0.172 8	0.152 1	0.139 6	0.172 3	0.149 5	0.137 7	0.163 3	0.135 4
ant-1.4	0.135 1	0.153 2	0.129 6	0.146 7	0.144 7	0.144 7	0.118 5	0.143 2
ant-1.5	0.072 5	0.069 5	0.056 4	0.073 7	0.066 0	0.067 8	0.072 5	0.066 0
ant-1.6	0.282 9	0.270 2	0.359 2	0.272 0	0.317 3	0.269 7	0.276 6	0.273 3
ant-1.7	0.204 3	0.383 4	0.275 1	0.225 9	0.286 9	0.284 4	0.253 5	0.226 8
camel-1.0	0.022 6	0.022 6	0.022 6	0.022 6	0.022 6	0.022 6	0.022 6	0.022 6
camel-1.2	0.619 4	0.650 3	0.767 0	0.655 4	0.677 7	0.635 2	0.624 1	0.597 7
camel-1.4	0.255 2	0.287 6	0.280 3	0.240 2	0.272 6	0.287 2	0.265 1	0.243 2
camel-1.6	0.396 2	0.375 6	0.369 1	0.388 4	0.370 7	0.387 6	0.387 8	0.368 3
xerces-1.2	0.122 5	0.096 4	0.098 7	0.122 5	0.105 5	0.107 7	0.112 7	0.118 0
xerces-1.3	0.313 3	0.296 4	0.301 8	0.317 1	0.298 5	0.269 7	0.365 4	0.314 9
ivy-1.1	0.657 1	0.693 5	0.883 9	0.665 1	0.610 6	0.844 5	0.665 1	0.631 6
ivy-1.4	0.042 4	0.033 0	0.045 5	0.042 4	0.038 2	0.042 4	0.042 4	0.050 7
ivy-2.0	0.099 8	0.099 1	0.063 4	0.104 3	0.103 8	0.092 0	0.095 9	0.092 6
mean	0.242 6	0.255 9	0.270 9	0.246 3	0.247 5	0.255 8	0.247 5	**0.234 6**
Win/Draw/Loss	9/1/4	9/1/4	7/1/6	9/1/4	8/2/4	8/1/5	10/1/3	

信息增益、卡方检验进行比较，“Win/Draw/Loss”的结果分别为“10/1/3”“10/1/3”“10/2/2”。MIC-TSFS 与 HFSNFP 方法、FSCR 以及 FSNNP 对比，“Win/Draw/Loss”的结果为“10/1/3”“9/2/3”“9/1/4”，AAE 分别降低了 5.9%、10.5%、3.8%。倒数第二行 mean 表示在所有数据集上 AAE，贝叶斯岭回归模型下在 MIC-TSFS 上，AAE 的均值最低，即在所有数据集上，MIC-TSFS 方法的整体平均绝对误差最小。

表 3 列出了 MIC-TSFS 方法、全特征（full）、ReliefF 算法、信息增益、卡方检验、FSCR、FSDNP 以及 HFSNFP 方法等在贝叶斯岭归模型中的 ARE

值。MIC-TSFS 与这几种方法对比的“Win/Draw/Loss”的结果分别为“9/1/4”“9/1/4”“7/1/6”“9/1/4”“8/2/4”“8/1/5”“10/1/3”。在大多数的数据集下，MIC-TSFS 能获得较优的结果，而且在所有数据集上 MIC-TSFS 的 ARE 最低。由此可见在贝叶斯岭回归下 MIC-TSFS 的性能相对其他特征方法的性能较好，即选出的特征能构造更有效的贝叶斯岭回归缺陷数目预测模型。

在线性回归模型中，各种方法的 AAE 与 ARE 的对比结果见表 4 和表 5。在表 4 中，MIC-TSFS 方法与全特征（full）、ReliefF 算法、信息增益、卡方检验、HFSNFP 方法、FSCR 以及 FSDNP 中 AAE 的“Win/Draw/Loss”对比结果分别为：“8/2/4”“8/2/4”“9/1/4”“9/1/4”“8/1/5”“8/1/5”“8/1/5”。对比 FSCR 以及 FSDNP 方法，AAE 分别降低 3.9%、1.5%。在大多数数据集中，MIC-TSFS 方法对比其他特征选择方法，得到的 AAE 较小。

在表 5 中，在线性回归模型下，MIC-TSFS 方法对比全特征、卡方检验的“Win/Draw/Loss”对比结果分别为“8/2/4”“9/1/4”，表明在大部分数据集中，MIC-TSFS 方法得到的 ARE 较小；MIC-TSFS 与 ReliefF 算法、信息增益、HFSNFP 方法、FSCR 方法以及 FSDNP 方法的 ARE “Win/Draw/Loss”对比结果分别为“7/2/5”“7/1/6”“7/1/6”“7/1/6”“7/1/6”，表明在过半的数据集上 MIC-TSFS 比 ReliefF 算法、信息增益、HFSNFP 方法、FSCR 方法以及 FSDNP 方法，MIC-TSFS 所得到的 ARE 小，且 MIC-TSFS 方法选出的特征构建的线性回归模型在所有数据集上的 AAE 与 ARE 最低。因此 MIC-TSFS 方法选出的特征也能构造更为有效的线性回归预测模型。

表 6 和表 7 分别列出了在决策树回归模型下各种方法的 AAE 与 ARE 的对比结果。

在表 6 中，MIC-TSFS 方法所有数据集的 AAE 均值最小，且 MIC-TSFS 方法与全特征、ReliefF 算法、信息增益、卡方检验、HFSNFP 方法的 AAE “Win/Draw/Loss”比较结果分别为：“9/1/4”“9/1/4”“8/2/4”“8/2/4”“8/1/5”。在表 7 中，所有数据集上 MIC-TSFS 的 ARE 均值最低，且 MIC-TSFS 方法与全特征、ReliefF 算法、信息增益、卡方检验、HFSNFP 方法的 AAE“Win/Draw/Loss”的结果分别为“9/1/4 ”“8/1/5”“9/2/3”“8/2/4”“8/2/4”。

表 4　线性回归模型的 AAE

	full	ReliefF	info	chi2	HFSNFP	FSCR	FSDNP	MIC-TSFS
ant-1.3	0.310 3	0.304 5	0.310 9	0.326 9	0.280 1	0.313 2	0.255 1	0.303 8
ant-1.4	0.314 7	0.286 9	0.275 8	0.264 7	0.275 8	0.264 1	0.264 7	0.297 4
ant-1.5	0.109 1	0.136 7	0.123 1	0.119 8	0.109 4	0.126 4	0.130 1	0.122 8
ant-1.6	0.484 4	0.459 0	0.737 9	0.635 9	0.547 4	0.464 4	0.469 9	0.458 7
ant-1.7	0.398 8	0.595 7	0.585 3	0.598 6	0.449 7	0.448 3	0.421 5	0.418 9
camel-1.0	0.044 2	0.044 2	0.044 2	0.044 2	0.044 2	0.044 2	0.044 2	0.044 2
camel-1.2	1.034 0	1.047 5	1.165 6	1.070 1	1.088 0	1.042 1	1.057 0	1.065 2
camel-1.4	0.496 7	0.506 9	0.544 7	0.596 3	0.495 0	0.508 6	0.515 1	0.494 4
camel-1.6	0.708 5	0.705 3	0.718 1	0.811 0	0.708 4	0.714 4	0.704 5	0.700 2
xerces-1.2	0.309 1	0.261 4	0.259 1	0.261 4	0.270 5	0.275 0	0.272 7	0.302 3
xerces-1.3	0.574 0	0.524 4	0.560 2	0.608 4	0.544 2	0.555 6	0.630 8	0.502 8
ivy-1.1	1.548 5	1.331 4	1.976 5	2.066 7	1.354 5	1.547 7	1.356 8	1.321 2
ivy-1.4	0.078 5	0.070 2	0.078 5	0.074 3	0.078 5	0.074 3	0.082 7	0.091 0
ivy-2.0	0.170 3	0.170 3	0.170 2	0.181 7	0.178 7	0.167 5	0.184 6	0.170 3
mean	0.470 1	0.460 3	0.539 3	0.547 1	0.458 9	0.467 6	0.456 4	**0.449 5**
Win/Draw/Loss	8/2/4	8/2/4	9/1/4	9/1/4	8/1/5	8/1/5	8/1/5	

表 5　线性回归模型的 ARE

	full	ReliefF	info	chi2	HFSNFP	FSCR	FSDNP	MIC-TSFS
ant-1.3	0.185 2	0.179 0	0.157 7	0.182 1	0.158 5	0.176 0	0.134 8	0.178 2
ant-1.4	0.201 8	0.162 8	0.132 4	0.118 5	0.147 3	0.151 3	0.118 5	0.187 4
ant-1.5	0.069 2	0.081 3	0.089 8	0.056 4	0.061 0	0.082 9	0.071 4	0.076 0
ant-1.6	0.286 3	0.273 4	0.490 1	0.412 9	0.329 2	0.274 4	0.279 9	0.269 1
ant-1.7	0.220 6	0.386 5	0.327 2	0.392 1	0.271 3	0.274 4	0.250 8	0.247 1
camel-1.0	0.022 6	0.022 6	0.022 6	0.022 6	0.022 6	0.022 6	0.022 6	0.022 6
camel-1.2	0.620 2	0.6369	0.747 2	0.634 4	0.677 7	0.632 2	0.689 7	0.631 6
camel-1.4	0.277 0	0.288 1	0.306 6	0.381 8	0.296 2	0.282 5	0.284 0	0.276 7
camel-1.6	0.412 5	0.379 7	0.395 7	0.497 4	0.389 2	0.390 7	0.393 0	0.386 1
xerces-1.2	0.150 9	0.096 4	0.098 3	0.096 4	0.105 5	0.114 5	0.116 4	0.145 6
xerces-1.3	0.345 0	0.266 6	0.293 6	0.358 5	0.299 0	0.289 0	0.403 7	0.298 8
ivy-1.1	0.695 0	0.624 8	0.922 0	0.997 1	0.603 8	0.788 9	0.674 1	0.610 4
ivy-1.4	0.042 4	0.033 0	0.041 3	0.034 0	0.038 2	0.041 3	0.042 4	0.052 8
ivy-2.0	0.102 8	0.102 8	0.074 8	0.116 2	0.104 0	0.094 8	0.093 1	0.102 8
mean	0.259 4	0.252 4	0.292 8	0.307 2	0.250 3	0.258 3	0.251 7	**0.248 9**
Win/Draw/Loss	8/2/4	7/2/5	7/1/6	9/1/4	7/1/6	7/1/6	7/1/6	

表 6　决策树回归模型的 AAE

	full	ReliefF	info	chi2	HFSNFP	FSCR	FSDNP	MIC-TSFS
ant-1.3	0.337 4	0.328 8	0.382 3	0.349 1	0.336 5	0.340 9	0.280 8	0.320 5
ant-1.4	0.367 1	0.314 4	0.363 2	0.308 5	0.336 9	0.319 9	0.376 1	0.352 0
ant-1.5	0.112 5	0.129 7	0.109 2	0.150 9	0.136 4	0.129 7	0.157 0	0.119 4
ant-1.6	0.488 8	0.495 6	0.470 2	0.639 4	0.524 2	0.495 6	0.470 5	0.452 9
ant-1.7	0.459 7	0.537 8	0.470 1	0.653 6	0.447 0	0.435 7	0.422 4	0.437 8
camel-1.0	0.0413	0.041 3	0.041 3	0.041 3	0.041 3	0.041 3	0.041 3	0.041 3
camel-1.2	1.027 5	1.098 1	1.064 3	1.157 3	1.147 3	1.079 9	1.121 0	1.020 7
camel-1.4	0.466 7	0.435 2	0.480 5	0.526 3	0.441 5	0.466 5	0.469 2	0.462 9
camel-1.6	0.689 6	0.663 9	0.694 3	0.694 9	0.658 0	0.661 4	0.691 7	0.731 6
xerces-1.2	0.313 6	0.322 7	0.306 8	0.306 8	0.311 4	0.356 4	0.350 0	0.354 5
xerces-1.3	0.463 3	0.483 7	0.558 3	0.528 4	0.526 4	0.524 1	0.464 8	0.464 3
ivy-1.1	1.973 8	2.012 7	1.972 7	2.048 5	1.905 3	2.054 5	2.045 5	1.891 7
ivy-1.4	0.111 8	0.078 3	0.103 5	0.103 5	0.099 2	0.103 5	0.103 5	0.103 5
ivy-2.0	0.204 3	0.202 9	0.187 3	0.187 9	0.210 4	0.195 9	0.190 2	0.198 9
mean	0.504 1	0.510 4	0.514 6	0.549 7	0.508 7	0.514 7	0.513 1	0.496 6
Win/Draw/Loss	9/1/4	9/1/4	8/2/4	8/2/4	8/1/5	8/2/4	7/2/5	

表 7　决策树回归模型的 ARE

	full	ReliefF	info	chi2	HFSNFP	FSCR	FSDNP	MIC-TSFS
ant-1.3	0.201 4	0.199 4	0.254 4	0.222 5	0.219 1	0.203 8	0.159 0	0.194 1
ant-1.4	0.248 2	0.195 8	0.235 3	0.191 3	0.230 7	0.206 9	0.247 2	0.230 7
ant-1.5	0.062 9	0.086 9	0.061 3	0.089 2	0.087 4	0.086 9	0.097 1	0.071 6
ant-1.6	0.255 2	0.281 3	0.249 3	0.366 6	0.284 7	0.287 0	0.270 7	0.223 6
ant-1.7	0.246 7	0.297 9	0.257 3	0.412 0	0.244 2	0.239 4	0.225 4	0.224 4
camel-1.0	0.019 7	0.019 7	0.019 7	0.019 7	0.019 7	0.019 7	0.019 7	0.019 7
camel-1.2	0.588 9	0.663 8	0.596 9	0.732 5	0.709 6	0.650 3	0.684 4	0.590 8
camel-1.4	0.255 1	0.217 2	0.253 2	0.280 7	0.226 1	0.244 4	0.256 8	0.233 5
camel-1.6	0.343 3	0.315 4	0.354 9	0.358 9	0.310 3	0.322 5	0.338 8	0.383 5
xerces-1.2	0.194 8	0.191 1	0.173 6	0.155 2	0.183 9	0.199 8	0.207 8	0.219 8
xerces-1.3	0.244 9	0.243 1	0.296 9	0.253 0	0.280 6	0.262 7	0.230 4	0.242 6
ivy-1.1	0.805 8	0.789 8	0.791 2	0.841 9	0.781 9	0.795 0	0.855 6	0.742 5
ivy-1.4	0.071 5	0.038 0	0.063 2	0.063 2	0.058 9	0.063 2	0.063 2	0.063 2
ivy-2.0	0.132 9	0.133 9	0.110 9	0.106 5	0.131 7	0.124 3	0.112 5	0.126 3
mean	0.262 2	0.262 4	0.265 6	0.292 4	0.269 2	0.264 7	0.269 2	0.254 7
Win/Draw/Loss	9/1/4	8/1/5	9/2/3	8/2/4	8/2/4	8/2/4	7/2/5	

说明在大多数数据集中，由 MIC-TSFS 方法所挑选的特征训练出的模型误差较小。MIC-TSFS 相比于 FSCR 与 FSDNP 方法，AAE 分别降低 3.5%、3.2%，ARE 分别降低 3.8%、5.4%。由此可以看出 MIC-TSFS 方法选择的特征，也能构造更为有效的决策树回归模型。

为进一步说明MIC-TSFS方法的有效性，表8～表10分别给出经由各种特征选择方法选出的特征子集分别在贝叶斯岭回归、线性回归以及决策树回归模型中得到 G-mean 值。

在表 8 中，MIC-TSFS 的 G-mean 均值最高。与 ReliefF 算法、信息增益、卡方检验、HFSNFP、FSCR 以及 FSDNP 的特征选择方法相比，G-mean 均值分别提升 4.9%、7.2%、4.1%、5.9%、9.8%、10.5%。

在表 9 中，MIC-TSFS 的 G-mean 同样具有最高的均值，与不使用特征选择方法的 full 相比，其 G-mean 均值提高 5.3%，而对比 ReliefF 算法、信息增益、卡方检验，G-mean 均值分别提升 6.4%、29.1%、8.2%。MIC-TSFS 相对于 HFSNFP、FSCR 以及 FSDNP 特征选择方法 G-mean 均值提升 3.7%、5.4%、13.5%。

从表 10 可以看出 MIC-TSFS 同样具有较高的 G-mean 均值。与 ReliefF 算法、信息增益、卡方检验、HFSNFP、FSCR 以及 FSDNP 的特征选择方法相比，G-mean 均值分别提升 3.2%、17.6%、2.8%、3.9%、3.2%、5.7%；而与不使用任何特征选择方法相比，G-mean 均值提升 3.8%。

在贝叶斯岭回归、线性回归以及决策树回归 3 种模型中，MIC-TSFS 方法的 G-mean 的平均值都达到最优，说明就整体情况而言，MIC-TSFS 方法要优于其他特征选择方法。然而，在数据集 camel-1.0 与 ivy-1.4 上，特征选择方法的大多数 G-mean 值为 0，其可能原因是这两个数据集中数据不平衡率较高（其中缺陷数目大于 0 的样

表 8　贝叶斯岭回归模型的 G-mean

	full	ReliefF	info	chi2	HFSNFP	FSCR	FSDNP	MIC-TSFS
ant-1.3	0.460 8	0.509 9	0.430 0	0.535 4	0.601 0	0.393 9	0.486 8	0.460 6
ant-1.4	0.183 2	0.259 1	0.259 1	0.172 5	0.213 1	0.330 7	0.235 0	0.237 2
ant-1.5	0.324 2	0.251 0	0.251 0	0.433 6	0.274 0	0.267 9	0.057 7	0.312 7
ant-1.6	0.778 2	0.781 5	0.792 5	0.784 5	0.730 9	0.771 6	0.773 5	0.787 4
ant-1.7	0.769 4	0.666 8	0.653 1	0.731 4	0.737 9	0.739 0	0.750 9	0.746 5
camel-1.0	0	0	0	0	0	0	0	0
camel-1.2	0.539 8	0.426 6	0.521 2	0.468 3	0.291 7	0.390 7	0.527 3	0.538 8
camel-1.4	0.588 8	0.567 5	0.606 8	0.572 8	0.515 9	0.627 5	0.579 2	0.568 9
camel-1.6	0.598 5	0.579 0	0.572 6	0.594 9	0.564 1	0.454 4	0.596 8	0.599 3
xerces-1.2	0.123 9	0.140 0	0.058 0	0.123 9	0.083 0	0.116 3	0.074 4	0.124 5
xerces-1.3	0.636 4	0.757 6	0.6052	0.539 2	0.759 1	0.641 4	0.564 3	0.647 5
ivy-1.1	0.333 2	0.305 3	0.396　7	0.292 3	0.477 5	0.323 6	0.292 3	0.514 9
ivy-1.4	0	0	0	0	0	0	0	0
ivy-2.0	0.372 6	0.364 1	0.345 2	0.405 4	0.311 3	0.300 4	0.385 8	0.348 5
mean	0.407 8	0.400 6	0.392 2	0.403 9	0.397 1	0.382 7	0.380 3	0.420 5

表 9　线性回归模型的 G-mean

	full	ReliefF	info	chi2	HFSNFP	FSCR	FSDNP	MIC-TSFS
ant-1.3	0.487 0	0.626 2	0.297 8	0.373 8	0.769 9	0.493 9	0.653 7	0.532 0
ant-1.4	0.397 3	0.290 8	0.057 7	0.407 7	0.261 2	0.412 2	0	0.425 9
ant-1.5	0.319 1	0.197 6	0.496 1	0.422 7	0.348 7	0.239 2	0.098 6	0.376 7
ant-1.6	0.753 9	0.782 9	0.594 0	0.689 2	0.720 5	0.769 9	0.773 5	0.784 7
ant-1.7	0.633 5	0.6765	0.552 9	0.691 3	0.737 6	0.748 3	0.752 8	0.739 2
camel-1.0	0	0	0	0	0	0	0	0
camel-1.2	0.548 4	0.488 8	0.284 3	0.446 7	0.373 6	0.444 9	0.525 7	0.529 3
camel-1.4	0.543 1	0.584 7	0.517 5	0.606 5	0.637 7	0.636 7	0.589 6	0.6
camel-1.6	0.626 2	0.587 0	0.578 8	0.559 1	0.583 4	0.558 6	0.593 4	0.602 5
xerces-1.2	0.176 0	0.110 2	0.113 6	0.112 2	0.123 7	0.115 7	0.187 1	0.188 5
xerces-1.3	0.635 2	0.726 4	0.522 6	0.705 0	0.749 2	0.624 5	0.647 9	0.662 8
ivy-1.1	0.438 6	0.614 9	0.589 9	0.507 1	0.652 0	0.518 9	0.484 3	0.637 9
ivy-1.4	0.168 5	0.070 7	0.070 7	0	0	0.070 7	0	0.070 7
ivy-2.0	0.409 4	0.318 4	0.328 9	0.450 7	0.277 0	0.499 0	0.385 8	0.313 6
mean	0.438 3	0.433 9	0.357 5	0.426 6	0.445 3	0.438 0	0.406 6	0.461 7

表 10　决策树回归模型的 G-mean

	full	ReliefF	info	chi2	HFSNFP	FSCR	FSDNP	MIC-TSFS
ant-1.3	0.545 7	0.491 8	0.533 0	0.536 2	0.511 5	0.425 5	0.573 8	0.547 5
ant-1.4	0.272 5	0.369 3	0.320 4	0.235 1	0.284 7	0.352 4	0.214 5	0.292 9
ant-1.5	0.360 4	0.292 6	0.055 6	0.347 6	0.319 4	0.292 6	0.112 2	0.330 0
ant-1.6	0.727 7	0.738 1	0.574 8	0.668 9	0.683 8	0.733 1	0.739 9	0.729 2
ant-1.7	0.676 8	0.636 8	0.573 7	0.652 2	0.690 5	0.672 1	0.669 7	0.662 5
camel-1.0	0	0	0	0	0	0	0	0
camel-1.2	0.532 0	0.497 1	0.463 2	0.490 5	0.473 1	0.528 5	0.475 6	0.517 8
camel-1.4	0.583 1	0.533 4	0.401 1	0.492 4	0.552 5	0.379 9	0.530 5	0.497 9
camel-1.6	0.474 2	0.484 5	0.5	0.505 3	0.512 8	0.508 5	0.453 9	0.486 3
xerces-1.2	0.372 7	0.389 3	0.256 7	0.503 1	0.543 4	0.432 1	0.409 3	0.46 9
xerces-1.3	0.452 1	0.408 8	0.577 3	0.590 2	0.454 8	0.601 9	0.593 6	0.592 6
ivy-1.1	0.515 5	0.582 6	0.600 7	0.536 9	0.499 9	0.490 2	0.553 3	0.597 6
ivy-1.4	0	0	0	0	0	0	0	0
ivy-2.0	0.281 8	0.402 6	0.257 9	0.289 0	0.264 0	0.412 6	0.365 7	0.290 4
mean	0.413 9	0.416 2	0.365 3	0.417 7	0.413 6	0.416 4	0.406 6	0.429 6

本的占比分别是 3.8%、6.7%），受数据不平衡的影响，这 3 种回归模型均没有正确找到的缺陷数不为 0 的样本，从而导致混淆矩阵中 TP 为 0。因此，在后续工作中可以考虑对不平衡因素进行处理，如结合 smote 过采样或代价敏感，以进一步提高模型预测性能。

对比在贝叶斯岭回归、线性回归、决策树回归模型下，MIC-TSFS 方法相对传统的特征选择方法的性能较好，这是因为 MIC-TSFS 除了考虑特征和标签之间相关性外，还考虑了特征与特征之间的冗余，通过特征聚类让较为相关的特征聚在同一个簇中，这样再从不同的簇中分别挑选特征，就能很好地改善特征冗余带来的负面影响，从而提高预测性能。与 HFSNFP、FSCR、FSDNP 方法相比，Pearson 系数仅能衡量特征间的线性关系，MIC-TSFS 方法采用最大信息系数，不仅可以描述线性关系，还可以很好地描述非线性、非函数关系等依赖关系，能够挖掘特征间以及特征与软件缺陷数目间更多的依赖关系。

从特征间的冗余性角度，MIC-TSFS 方法能够更为公平地删除更多的冗余特征，使得特征间的冗余性进一步降低。从特征与软件缺陷数目间的相关性角度，MIC-TSFS 方法能够最大限度地保留与软件缺陷数目相关的、更有区分能力的特征，也可以避免特征与软件缺陷数目的相关性评价的偏置。由此，MIC-TSFS 方法选择的特征对软件缺陷数目的识别能力更强，且冗余度低，因此预测准确率有所提升。

5　结束语

本文提出一种基于最大信息系数的软件数目预测特征选择方法 MIC-TSFS，用于删除冗余特征和不相关特征。在 PROMISE 公开数据集上进行了大量的实验，实验结果表明 MIC-TSFS 能够有效地删除冗余特征和不相关特征，可以提高软件缺陷数目预测模型的预测精度。但该方法仍有一些后续工作值得扩展，包括：在冗余性分析阶段，探讨聚类中止条件的设置对 MIC-TSFS 方法是否存在影响；在相关性分析阶段，研究获取的特征数最优值的方法；讨论数

据不平衡率对预测结果所产生的影响，并结合 smote 过采样方法、代价敏感等不平衡处理方法，进一步提高模型预测性能。此外，实验中 MIC-TSFS 方法仅应用于中小型项目（程序模块在 1 000 以内），讨论其在大型程序项目中的实际应用效果是下一步工作的重点。

参考文献：

[1] 宫丽娜, 姜淑娟, 姜丽. 软件缺陷预测技术研究进展[J]. 软件学报, 2019, 30(10): 3090-3114.
GONG L N, JIANG S J, JIANG L. Research progress of software defect prediction[J]. Journal of Software, 2019, 30(10): 3090-3114.

[2] 刘望舒, 陈翔, 顾庆, 等. 软件缺陷预测中基于聚类分析的特征选择方法[J]. 中国科学: 信息科学, 2016, 46(9): 1298.
LIU W S, CHEN X, GU Q, et al. A cluster-analysis-based feature-selection method for software defect prediction[J]. SCIENTIA SINICA Informationis, 2016, 46(9): 1298.

[3] GRAVES T L, KARR A F, MARRON J S, et al. Predicting fault incidence using software change history[J]. IEEE Transactions on Software Engineering, 2000, 26(7): 653-661.

[4] OSTRAND T J, WEYUKER E J, BELL R M. Predicting the location and number of faults in large software systems[J]. IEEE Transactions on Software Engineering, 2005, 31(4): 340-355.

[5] CHEN M, MA Y. An empirical study on predicting defect numbers[C]//Proceeding of the 27th International Conference on Software Engineering and Knowledge Engineering. [S.l.:s.n.], 2015: 397-402.

[6] RATHORE S S, KUMAR S. Predicting number of faults in software system using genetic programming[J]. Procedia Computer Science, 2015: 303-311.

[7] RATHORE S S, KUMAR S. A decision tree regression based approach for the number of software faults prediction[J]. ACM Sigsoft Software Engineering Notes, 2016, 41(1): 1-6.

[8] RATHORE S S, KUMAR S. An empirical study of some software fault prediction techniques for the number of faults prediction[J]. Soft Computing, 2017, 21(24): 7417-7434.

[9] CHEN X, ZHANG D, ZHAO Y, et al. Software defect number prediction: unsupervised vs supervised methods[J]. Information and Software Technology, 2019(106): 161-181.

[10] RATHORE S S, KUMAR S. Towards an ensemble based system for predicting the number of software faults[J]. Expert Systems with Applications, 2017(82): 357-382.

[11] RATHORE S S, KUMAR S. Linear and non-linear heterogeneous ensemble methods to predict the number of faults in software systems[J]. Knowledge-Based Systems, 2017(119): 232-256.

[12] YU X, LIU J, YANG Z, et al. Learning from imbalanced data for predicting the number of software defects[C]//Proceeding of the International Symposium on Software Reliability Engineering. [S.l.:s.n.], 2017: 78-89.

[13] 刘洺辛, 陈晶, 王麒媛. 基于改进特征选择方法的文本情感分类研究[J]. 电信科学, 2018, 34(10): 85-95.
LIU M X, CHEN J, WANG Q Y. Research on text sentiment classification based on improved feature selection method[J]. Telecommunications Science, 2018, 34(10): 85-95.

[14] 李叶飞, 官国飞, 葛崇慧. FSDNP: 针对软件缺陷数预测的特征选择方法[J]. 计算机工程与应用, 2019, 55(14): 61-68.
LI Y F, GUAN G F, GE C H. FSDNP: feature selection method for software defect number prediction[J]. Computer Engineering and Applications, 2019, 55(14): 61-68.

[15] YU X, MA Z, MA C, et al. FSCR: a feature selection method for software defect prediction[C]//Proceeding of the 29th International Conference on Software Engineering and Knowledge Engineering. [S.l.:s.n.], 2017: 351-356.

[16] 马子逸, 马传香, 刘瑞奇. 面向软件缺陷个数预测的混合式特征选择方法[J]. 计算机应用研究, 2018, 35(2): 487-502.
MA Z Y, MA C X, LIU R Q. Hybrid feature selection method for number of software faults prediction[J]. Application Research of Computers, 2018, 35(2): 487-502.

[17] RESHEF D N, RESHEF Y A, FINUCANE H K, et al. Detecting novel associations in large data sets[J]. Science, 2011, 334(6062): 1518-1524.

[18] GAO K, KHOSHGOFTAAR T M, WANG H, et al. An empirical investigation of filter attribute selection techniques for software quality classification[C]//Proceeding of the 10th IEEE international conference on Information Reuse & Integration. Piscataway: IEEE Press, 2009: 272-277.

[19] JURECZKO M. Signifificance of different software metrics in defect prediinformation reuse and integrationction[J]. Software Engineering, 2011, 1(1): 86-95.

[作者简介]

刘国庆（1995− ），男，杭州电子科技大学计算机学院硕士生，主要研究方向为软件测试、机器学习。

王兴起（1974− ），男 ，博士，杭州电子科技大学教授，主要研究方向为数据挖掘、软件测试等。

魏丹（1979− ），女，博士，杭州电子科技大学讲师，主要研究方向为数据挖掘、智能软件工程等。

方景龙（1964− ），男，博士，杭州电子科技大学教授，主要研究方向为智能软件工程、机器学习和人工智能等。

邵艳利（1989− ），女，博士，杭州电子科技大学副研究员，主要研究方向为 CAD&CAE、MBSE、智能软件工程和数据挖掘等。