(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 110471856 A
(43)申请公布日 2019.11.19

(21)申请号 201910775361.6

(22)申请日 2019.08.21

(71)申请人 大连海事大学
地址 116026 辽宁省大连市高新园区凌海路1号

(72)发明人 郭世凯 董剑 陈荣 王佳慧 李辉 郭晨 唐文君

(74)专利代理机构 大连东方专利代理有限责任公司 21212
代理人 姜玉蓉 李洪福

(51)Int.Cl.
G06F 11/36(2006.01)
G06K 9/62(2006.01)

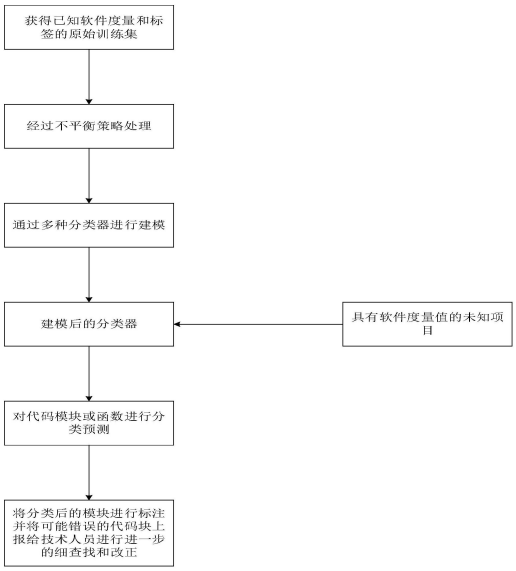
权利要求书1页 说明书3页 附图1页

(54)发明名称

一种基于数据不平衡的软件缺陷预测方法

(57)摘要

本发明公开了一种基于数据不平衡的软件缺陷预测方法，包括：从已知bug分布的项目中，将具有软件度量值的各类错误报告作为预测使用的原始数据集；采用RSMOTE不平衡处理策略对原数据集中文本矩阵进行不平衡处理、得到平衡数据集；使用朴素贝叶斯、多项式朴素贝叶斯、K近邻、支持向量机、分类树、和Adacost对平衡数据集进行建模找到预测效果最佳的分类器；提取未知bug位置的新项目的软件度量值，输入分类器，进行预测，输出每个程序段是否有bug的预测信息，并进行记录存储。本方法采用了RSMOTE不平衡处理策略对原数据集中文本矩阵进行不平衡处理，因此生成少数类样例更加灵活，能够产生更加广泛合理的样例。



CN 110471856 A

1.一种基于数据不平衡的软件缺陷预测方法,其特征在于包括:

将具有软件度量值的各类错误报告作为预测使用的原始数据集;

采用RSMOTE不平衡处理策略对原数据集中文本矩阵进行不平衡处理、得到平衡数据集;

使用朴素贝叶斯、多项式朴素贝叶斯、K近邻、支持向量机、分类树、和Adacost对平衡数据集进行建模找到预测效果最佳的分类器;

将新错误报告输入至分类器内对语句块进行分类、找到出现错误的语句块实现对bug位置的预测。

2.根据权利要求1所述的方法,其特征还在于:所述对原数据集中文本矩阵进行不平衡处理具体采用采用如下方式:

S21:将原始数据集中少数类进行标记,进行不平衡倍率N的计算;

S22:选择一个少数类样本点X、并对该少数类样本点的k个少数类近邻进行记录;

S23:按照欧式距离排序、依次选择一个近邻点Y;

S24:在以近邻点Y和少数类样本点X的欧式距离为直径的圆内生成新的少数类样本点;

S24:判断生成样本点是否在所选两个样本点形成的圆内、以及生成样本点的欧式距离最近的样本点是否为少数类,若不是少数类或不在所选两个样本点形成的圆内则返回S23重新生成样本点,若满足条件则继续下一步操作;

S25:判断该样本点是否需要继续生成少数类样本点,若选择的少数类样本点X已生成的样本个数达到不平衡度N,则返回S22进行下一个少数类样本点的选择,若没有达到不平衡度N,则返回S23再选择下一个近邻点Y;

S26:当所有少数类样本点全部遍历完成,将生成的少数类样本加入原数据集以生成平衡数据集。

一种基于数据不平衡的软件缺陷预测方法

技术领域

[0001] 本发明涉及软件缺陷预测领域，尤其涉及一种基于数据不平衡的软件缺陷预测方法。

背景技术

[0002] 随着人们对软件需求的不断增加，软件开发也越来越重要，其中程序调试是软件开发过程中十分重要的一环，该过程主要包括故障检测、故障定位以及故障修复环节，其中故障定位是最为繁琐的环节。软件开发过程中，难免会出现一系列的故障，其中一部分可根据编译信息查找到并进行改正，但大部分程序故障是由逻辑错误所导致的，据统计修复软件中故障的成本占整个软件维护总成本的50％～80％，并且在修复过程中，需要有经验和对代码语义、结构等有了解的人员来完成。为了解决上述问题，研究人员提出了多种软件度量，提出的软件度量与结构和数据有关，如类内方法的数量、继承树的深度、直系继承类的数目、传出耦合和传入耦合类的个数、当该类的对象接收到消息时可以执行的不同方法的数量等等度量信息，结合这些软件度量值，通过分类器进行学习建模，将新项目与训练数据所建立模型进行比较，可以判断出某一代码段或函数段中是否存在错误，从而辅助开发人员快速找到程序故障所在，减轻程序人员手工排查错误的任务量，提高程序错误定位效率。然而在大部分工程中，故障程序段占总程序数目的少数，从而会产生类别不平衡现象，导致分类器的分类效果不好。因此许多学者针对此现象提出了多种不平衡处理方法，如基于数据层面的ROS、RUS、SMOTE方法以及基于算法层面的CSC方法，来对不平衡数据集进行处理，从而减小不平衡数据带来的影响

[0003] 但目前的不平衡处理方法显然是存在一些缺点的，如随机过采样采取简单复制样本的策略来增加少数类样本，这样容易产生模型过拟合的问题，即使得模型学习到的信息过于特别(Specific)而不够泛化(General)。随机欠采样则是由于采样的样本集合要少于原来的样本集合，因此会造成一些信息缺失，即将多数类样本删除有可能会导致分类器丢失有关多数类的重要信息。而按照SMOTE算法进行线性插值合成新的样例后，得到的新的少数类只能分布在原少数类实例之间的线段中，严格的限制了新生成的少数类实例的分布范围，算法层面的CSC也存在着一些局限性，如当正类过少的时候，即使将正类错分到负类的代价很大，它也仍然更倾向与将正类分到负类。

发明内容

[0004] 根据现有技术存在的问题，本发明公开了一种基于数据不平衡的软件缺陷预测方法，具体包括如下步骤：

[0005] 将具有软件度量值的各类错误报告作为预测使用的原始数据集；即提取已知bug分布的项目的软件度量值作为属性，是否存在bug作为标签，形成用于训练分类器的原始数据集。

[0006] 采用RSMOTE不平衡处理策略对原数据集中文本矩阵进行不平衡处理、得到平衡数

据集；

[0007] 使用朴素贝叶斯、多项式朴素贝叶斯、K近邻、支持向量机、分类树、和Adacost对平衡数据集进行建模找到预测效果最佳的分类器；

[0008] 提取未知bug位置的新项目的软件度量值，并输入到分类器，进行预测，输出是否存在bug的信息，并进行存储，以实现对bug位置的预测。

[0009] 进一步的，所述对原数据集中文本矩阵进行不平衡处理具体采用如下方式：

[0010] S21:将原始数据集中少数类进行标记，进行不平衡倍率N的计算；

[0011] S22:选择一个少数类样本点X、并对该少数类样本点的k个少数类近邻进行记录；

[0012] S23:按照欧式距离排序、依次选择一个近邻点Y；

[0013] S24:在以近邻点Y和少数类样本点X的欧式距离为直径的圆内生成新的少数类样本点；

[0014] S24:判断生成样本点是否在所选两个样本点形成的圆内、以及生成样本点的欧式距离最近的样本点是否为少数类，，若不是少数类或不在所选两个样本点形成的圆内则返回S23重新生成样本点，若满足条件则继续下一步操作；

[0015] S25:判断该样本点是否需要继续生成少数类样本点，若选择的少数类样本点X已生成的样本个数达到不平衡度N，则返回S22进行下一个少数类样本点的选择，若没有达到不平衡度N，则返回S23再选择下一个近邻点Y；

[0016] S26:当所有少数类样本点全部遍历完成，将生成的少数类样本加入原数据集以生成平衡数据集。

[0017] 由于采用了上述技术方案，本发明提供的一种基于数据不平衡的软件缺陷预测方法，由于本方法采用了RSMOTE不平衡处理策略对原数据集中文本矩阵进行不平衡处理，因此生成少数类样例更加灵活，能够产生更加广泛合理的样例，使数据更加均匀，更有利于提高分类的准确性。

附图说明

[0018] 为了更清楚地说明本申请实施例或现有技术中的技术方案，下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍，显而易见地，下面描述中的附图仅仅是本申请中记载的一些实施例，对于本领域普通技术人员来讲，在不付出创造性劳动的前提下，还可以根据这些附图获得其他的附图。

[0019] 图1为本发明方法的流程图。

具体实施方式

[0020] 为使本发明的技术方案和优点更加清楚，下面结合本发明实施例中的附图，对本发明实施例中的技术方案进行清楚完整的描述：

[0021] 如图1所示的一种基于数据不平衡的软件缺陷预测方法，具体包括如下步骤：

[0022] S1:将具有软件度量值的各类错误报告作为预测使用的原始数据集，即将已知bug分布的程序分为小的程序段，再从每个程序段中提取出各个软件度量值，如继承树深度、每一类中方法的个数、类的直系继承类的个数等，再根据是否存在bug来确定该程序段的标签，将各类软件度量值作为属性，是否有bug作为标签，形成原始数据集。

[0023] S2:采用RSMOTE不平衡处理策略对原数据集中文本矩阵进行不平衡处理、得到平衡数据集;

[0024] 算法具体内容如下:

[0025] 步骤一:少数类标记:将原始数据集中少数类进行标记,进行不平衡倍率N的计算;

[0026] 步骤二:少数类选择以及K近邻记录:选择一个少数类样本点X的并对该少数类样本点的k个少数类近邻进行记录。

[0027] 步骤三:近邻选择:按照欧式距离排序(可从大到小或从小到大),依次选择一个近邻Y。

[0028] 步骤四:生成新样本点X_{new}:依据生成函数在以Y和X的欧式距离为直径的圆内生成新的少数类样本点。

[0029] 步骤五:约束判断:主要判断生成样本点是否在所选两个样本点形成的圆内,以及生成样本点的欧式距离最近的样本点是否为少数类,若不是少数类则返回步骤三重新生成样本点,若是少数类则继续下一步操作。

[0030] 步骤六:单个样本点循环结束条件判断:判断该点是否需要继续生成少数类样本点,若选择的少数类样本点X已生成的样本数个数达到不平衡度N,则返回步骤二,进行下一个少数类样本点的选择,若没有达到N,则返回步骤三再选择下一个近邻Y。

[0031] 步骤七:全部循环结束条件判断:当所有少数类样本点全部遍历完成,将生成的少数类样本加入原数据集,以生成平衡的数据集。

[0032] 对于步骤四的生成函数定义如下:

[0033] 假设选择少数类样本X具有n个属性分别为$x_1,x_2,x_3,...,x_n$,选择的近邻Y具有n个属性分别为$y_1,y_2,y_3,...,y_n$,则新样本的产生空间为:

[0034] $z_{1i}=x_j-|y_{ij}-x_j|$,$z_{2i}=x_j+|y_{ij}-x_j|$ $i=1,2,...,N,j=1,2,...,n$

[0035] 生成函数为:

[0036] 与X_i按照如下公式构建一个新样本

[0037] $x_{new}=x_j+random(0,1)\times(z_{2i}-z_{1i})$ $i=1,2,...,N,j=1,2,...,n$

[0038] 对于步骤五的约束判断,主要有两点判断因素,一是找到距离新生成样本点最近邻的点的标签,查询是否与生成点相同,二是判断欧式距离是否满足条件,欧式距离判断可以表示为下式

[0039] 限制欧式距离:

[0040] $||X_{inew}-X||<||Y_i-X||$

[0041] S3:使用朴素贝叶斯、多项式朴素贝叶斯、K近邻、支持向量机、分类树、和Adacost,对经过不平衡处理后的数据集进行建模,找到预测效果最佳的分类器。

[0042] S4:将新错误报告输入至分类器内对语句块进行分类、找到出现错误的语句块实现对bug位置的预测。即使用与S1相同的方法提取未知bug位置的新项目的软件度量值,并输入到分类器,进行预测,输出每一样本即程序段内是否存在bug的信息,并将所有样本输出结果进行存储,以实现对bug位置的预测。

[0043] 以上所述,仅为本发明较佳的具体实施方式,但本发明的保护范围并不局限于此,任何熟悉本技术领域的技术人员在本发明揭露的技术范围内,根据本发明的技术方案及其发明构思加以等同替换或改变,都应涵盖在本发明的保护范围之内。

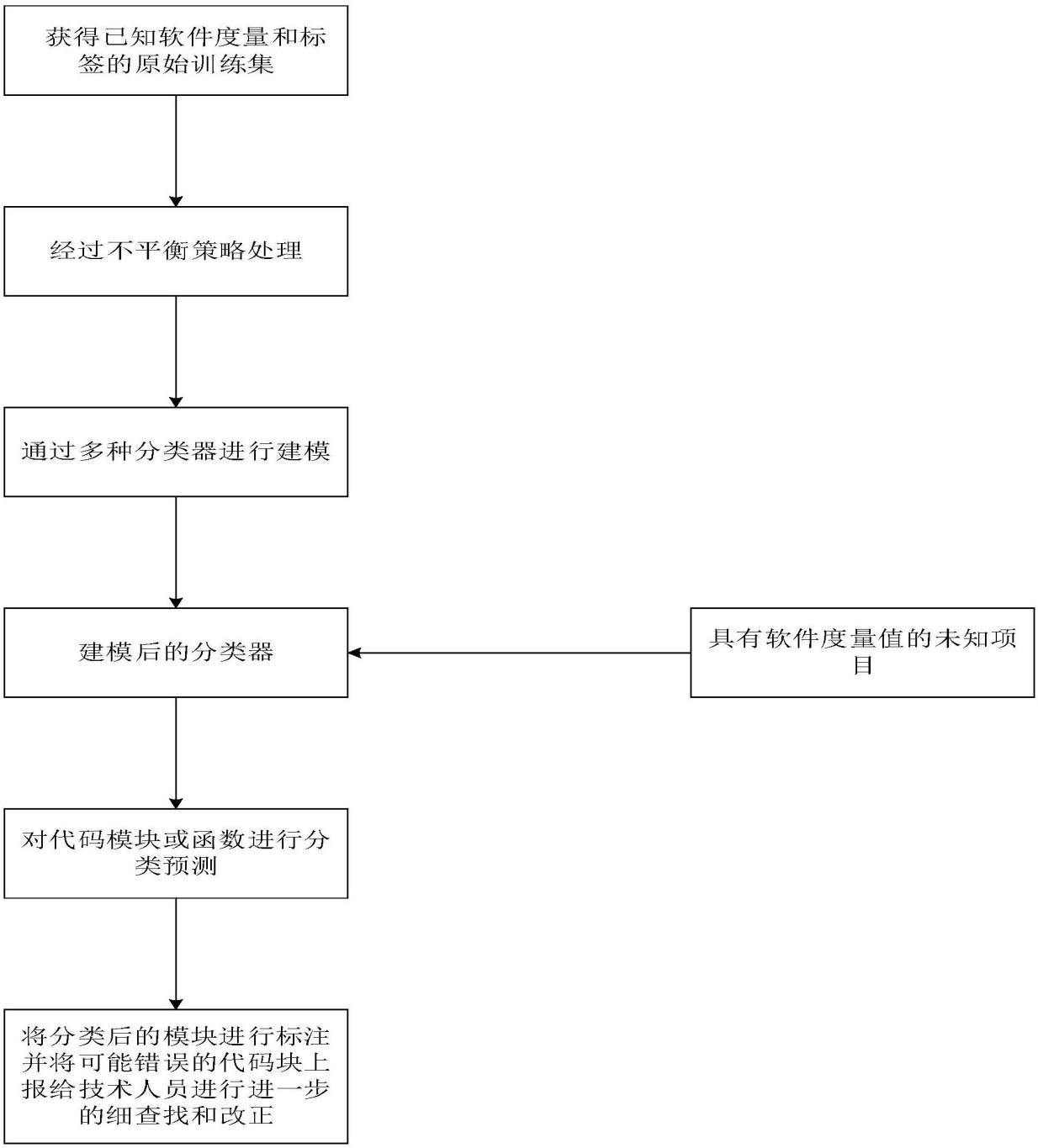

获得已知软件度量和标签的原始训练集
经过不平衡策略处理
通过多种分类器进行建模
建模后的分类器
具有软件度量值的未知项目
对代码模块或函数进行分类预测
将分类后的模块进行标注并将可能错误的代码块上报给技术人员进行进一步的细查找和改正

图1