(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 110597735 A
(43)申请公布日 2019.12.20

(21)申请号 201910907932.7

(22)申请日 2019.09.25

(71)申请人 北京航空航天大学
地址 100191 北京市海淀区学院路37号

(72)发明人 艾骏 王飞 许嘉熙 郭皓然 邹卓良 施韬

(74)专利代理机构 北京永创新实专利事务所 11121
代理人 祗志洁

(51)Int.Cl.
G06F 11/36(2006.01)
G06F 8/41(2018.01)
G06N 3/04(2006.01)
G06N 3/08(2006.01)
G06K 9/62(2006.01)

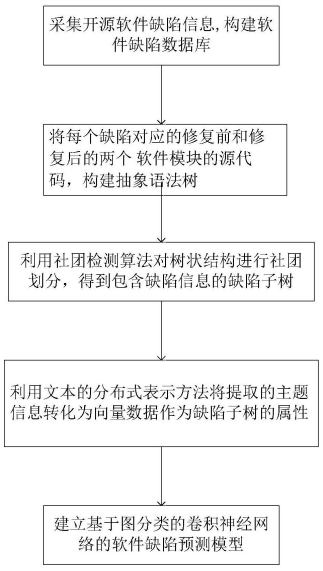
权利要求书2页 说明书6页 附图1页

(54)发明名称

一种面向开源软件缺陷特征深度学习的软件缺陷预测方法

(57)摘要

本发明提供了一种面向开源软件缺陷特征深度学习的软件缺陷预测方法，属于软件工程技术领域。本发明包括：采集开源软件缺陷信息，构建软件缺陷数据库，将源代码生成抽象语法树；利用社团检测算法将抽象语法树进行剪枝得到缺陷子树，然后结合修复描述和项目基础信息、源代码来建立缺陷子树的信息语料库，从中提取主题单词并转化为向量表示，作为缺陷子树中节点的属性；最后建立基于图分类的卷积神经网络的软件缺陷预测模型，将缺陷子树表示为邻接矩阵和属性矩阵作为模型的输入训练卷积神经网络，识别待预测软件模块源代码是否具有缺陷倾向性。本发明利用深度学习的方法直接从结构化的软件代码中提取缺陷深度特征，能够取得更好的缺陷识别效果。



CN 110597735 A

1.一种面向开源软件缺陷特征深度学习的软件缺陷预测方法,其特征在于,包括:

步骤1,采集开源软件缺陷信息,构建软件缺陷数据库,存储的每条缺陷信息包括软件模块在软件修复前和修复后两个版本的源代码,以及文本类型的修复描述和项目基础信息;

步骤2,依次取出软件缺陷数据库中每个缺陷对应的软件模块的修复前和修复后的两个版本的源代码,对应生成两个抽象语法树,对比两个抽象语法树结构,得到其中的差异节点,并为两个抽象语法树的各节点添加节点是否发生修改的属性;

步骤3,利用社团检测算法对每个缺陷对应的两个抽象语法树进行社团划分,将所有包含差异节点的社团进行最小化连接,得到包含缺陷信息的缺陷子树;对每个缺陷,利用相同社团检测算法得到软件修复前和修复后的缺陷子树;

步骤4,对每个缺陷子树,由对应的修复描述、项目基础信息以及缺陷子树的语义信息获取缺陷子树的信息语料库,利用主题建模技术从信息语料库中获取缺陷子树的主题单词,利用文本的分布式表示方法将主题单词转化为一维向量数据,并作为缺陷子树中节点的属性;

步骤5,建立基于图分类的卷积神经网络的软件缺陷预测模型,将步骤四中缺陷子树表示为邻接矩阵和属性矩阵作为模型的输入训练卷积神经网络,识别待预测软件模块源代码是否具有缺陷倾向性;

设缺陷子树中有N个节点,则邻接矩阵A为N×N的矩阵,若两个节点间存在连接则在邻接矩阵中对应的元素为1,否则为0;设节点属性数量为D,则属性矩阵X是一个N×D的矩阵,节点的属性包括:节点是否发生修改,对应主题单词的一维向量中的各元素。

2.根据权利要求1所述的方法,其特征在于,所述的每条缺陷信息中,修复描述记录缺陷的原因、种类和故障表现,项目基础信息记录对应的软件项目基础信息,包括名称、主题、描述和语言。

3.根据权利要求1所述的方法,其特征在于,所述的步骤2中,对比两个抽象语法树结构,是将两个抽象语法树的节点进行匹配,对相似节点建立映射关系,一个抽象语法树上的节点只能添加到一个映射节点对中,匹配成功的节点是未发生修改的非缺陷代码节点,无法匹配的节点是发生修改行为的缺陷代码节点,即差异节点,进一步将是否发生修改作为属性储存在抽象语法树的节点中。

4.根据权利要求1所述的方法,其特征在于,所述的步骤3中,将抽象语法树看作有向无环图,利用Louvain社团检测算法将对抽象语法树划分为多个子树,得到抽象语法树中各个节点的子树归属,查找抽象语法树中的所有差异节点的子树归属,得到所有包含差异节点的子树集合,通过子树根节点间的最短路径将子树集合进行最小化连接,得到对原抽象语法树进行剪枝后的缺陷子树。

5.根据权利要求1所述的方法,其特征在于,所述的步骤4中,缺陷子树的信息语料库包括:项目基础信息和修复描述构成的背景信息;从缺陷子树中提取的语义信息,包括所在类的类名、成员变量名、方法名、方法返回值、方法形参名、方法形参类型及其他变量名。

6.根据权利要求1或5所述的方法,其特征在于,所述的步骤4中,在对缺陷子树的信息语料库进行主题提取前先进行预处理,包括:

缺陷子树的信息语料库中存储的为文本信息,对文本信息先进行分词,包括:根据空

格、标点符号和段落的分割方式，将文本信息划分为单词组；对代码中复合单词形式的名称和类型，根据大写区分和下划线区分进行拆分；

过滤英语语言中的停用词，过滤编程语言中的关键字和无具体含义的编程词汇；

提取词干，将一个英文单词的不同表达还原为一个单词。

7.根据权利要求1所述的方法，其特征在于，所述的步骤4中，采用隐含狄利克雷分布LDA提取主题，设置取每个缺陷子树的主题数为1，LDA在该主题下取词分布结果，使用概率最高的5个单词代表该主题。

8.根据权利要求1所述的方法，其特征在于，所述的步骤4中，利用word2vec方法将主题的每个单词映射为向量表示，设为h维向量，设每个缺陷子树的主题单词有m个，h、m均为整数，则得到缺陷子树的语义特征矩阵$S_{m\times h}$；进一步，将矩阵$S_{m\times h}$压缩为一维向量，一维向量的第k个元素为矩阵$S_{m\times h}$的第k列的m个元素的均值，k＝1,2,…h。

9.根据权利要求1所述的方法，其特征在于，所述的步骤5中，建立基于图分类的卷积神经网络的软件缺陷预测模型，设卷积神经网络的总层数为L，其中从l层聚合到l+1层的传播模型$H^{(l+1)}$表示为：

$H^{(l+1)}=\sigma(AH^{(l)}W^{(l)})$，$l\in L$；

其中，$H^{(0)}=X$，X为缺陷子树的属性矩阵，$H^{(L)}$是第L层的输出；A为缺陷子树的邻接矩阵；σ是ReLU非线性激活函数；$W^{(l)}$是第l层的权值矩阵；

在经过L层卷积层后使用池化技术获得用于图分类的图聚合结果，通过全连接层综合卷积神经网络学习到的所有信息，最后使用Softmax分类器输出是否具有缺陷。

10.根据权利要求1所述的方法，其特征在于，所述的步骤5中，在得到训练好的软件缺陷预测模型后，将待预测的软件模块的源代码转化为树状结构，对源代码的语义信息和项目基础信息，进行主题提取，再利用文本的分布式表示方法将主题单词转化为一维向量，作为节点属性，输入训练好的软件缺陷预测模型，输出待预测的软件模块是否存在缺陷。

一种面向开源软件缺陷特征深度学习的软件缺陷预测方法

技术领域

[0001] 本发明属于软件工程领域，涉及一种面向开源软件的基于抽象语法树结构特征学习的软件缺陷预测方法。

背景技术

[0002] 随着软件系统的规模和复杂度的增长，软件缺陷也与日俱增，如何提高软件的质量并及早的识别、预测、修复软件的缺陷，已成为整个软件生命周期中必须始终关心和设法解决的问题。软件缺陷预测能够根据软件代码特征和历史缺陷信息，及早的识别出存在能影响软件可靠性的软件缺陷倾向的模块，从而充分利用有效的资源提高软件产品的质量与可靠性。

[0003] 机器学习技术的逐渐成熟使得基于统计学习的数据驱动型软件缺陷预测变得更为有效。大量实例的研究表明，利用机器学习中的关联规则、分类器算法、聚类算法等，取得了很好的缺陷预测效果，对于提高软件可靠性有着显著的作用。早期基于机器学习的软件缺陷预测主要是用项目历史数据作为训练，得到模型在用于该项目未来版本中的软件缺陷，这种预测方法被称为项目内缺陷预测。然而项目内缺陷预测的预测结果受项目历史数据数量的限制明显，在工程场景中往往很难获取充足的训练数据，因此无法通过改进模型来提高预测的效果。针对项目内缺陷预测中训练数据不足和获取困难的瓶颈，一些研究人员建议利用其他项目数据来训练模型进行预测，提出跨项目缺陷预测。跨项目缺陷预测核心思想是使用其他软件中分布相似的度量数据给出待预测软件中的缺陷模块，其难点在于选择能够在项目间迁移的数据和缺陷度量元。开源软件为跨项目缺陷预测带来了规模巨大的可用数据，目前Github用户总量达到3100万，项目总数超过了9600万，合并请求超过了2亿次，其中包含海量的软件缺陷修复数据，缺陷代码的大数据使建立复杂深层的学习模型成为可能。

[0004] 获取有效的软件缺陷相关的度量元是准确识别高风险模块的关键。传统度量元集中在基于软件规模等度量元的缺陷预测方面，研究缺陷和软件规模、复杂度、模块耦合性等属性之间的关系，以此预测软件可能存在的缺陷数量。比较有代表性的度量元包括代码行数、McCabe复杂性度量、Halstead程序复杂度度量等。随着面向对象方法的普及，越来越多的面向对象程序的度量元得到越来越多的应用。面向对象中最为典型的是CK度量元和MOOD度量元。但是目前的度量元主要是将软件程序元素离散化处理，由文件或类颗粒度上的统计结果设定特征指标，属于可认知的浅层特征，然而现实世界中的很多软件缺陷并不具备某些特定的线性特征，而是包含难以挖掘的深层特征，因此传统度量元在实际的软件工程项目中效果不太理想。

[0005] 最近部分研究人员将抽象语法树应用在软件缺陷预测中，将抽象语法树转化为数值序列，使用深度学习的方法获取缺陷的有效表征(参考文件1:Wang S,Liu T,Nam J,et al.Deep Semantic Feature Learning for Software Defect Prediction[J].IEEE Transactions on Software Engineering,2018,PP(99):1-1.参考文件2:Dam H K,Tran

T,Pham T T M,et al.Automatic feature learning for predicting vulnerable software components[J].IEEE Transactions on Software Engineering,2018,PP(99):1-1.)。然而数值序列是树状结构的一种有损转换,并且原始的代码模块中缺陷信息被大量的无关信息所掩埋,实际预测效果仍然有待提高。

发明内容

[0006] 由于目前软件缺陷度量元主要根据专家经验将软件程序元素离散化处理,在文件或类颗粒度上的统计结果设定特征指标,属于可认知的浅层特征,然而现实世界中的很多度量元与缺陷间并不具备简单线性关系,因此传统度量元在实际的软件工程项目中效果不太理想,对实际中面向开源软件的缺陷预测中并不能取得好的预测效果。为解决现有技术的不足,本发明提出了一种面向开源软件缺陷特征深度学习的软件缺陷预测方法。

[0007] 本发明所实现的面向开源软件缺陷特征深度学习的软件缺陷预测方法,包括:

[0008] 步骤1,采集开源软件缺陷信息,构建软件缺陷数据库,存储的每条缺陷信息包括软件模块在软件修复前和修复后两个版本的源代码,以及文本类型的修复描述和项目基础信息;

[0009] 步骤2,依次取出软件缺陷数据库中每个缺陷对应的软件模块的修复前和修复后的两个版本的源代码,对应生成两个抽象语法树,对比两个抽象语法树结构,得到其中的差异节点,并为两个抽象语法树的各节点添加节点是否发生修改的属性;

[0010] 步骤3,利用社团检测算法对每个缺陷对应的两个抽象语法树进行社团划分,将所有包含差异节点的社团进行最小化连接,得到包含缺陷信息的缺陷子树;对每个缺陷,利用相同社团检测算法得到软件修复前和修复后的缺陷子树;

[0011] 步骤4,对每个缺陷子树,由对应的修复描述、项目基础信息以及缺陷子树的语义信息获取缺陷子树的信息语料库,利用主题建模技术从信息语料库中获取缺陷子树的主题,利用文本的分布式表示方法将主题单词转化为一维向量数据,并作为缺陷子树中节点的属性;

[0012] 步骤5,建立基于图分类的卷积神经网络的软件缺陷预测模型,将步骤四中缺陷子树表示为邻接矩阵和属性矩阵作为模型的输入训练卷积神经网络,识别待预测软件模块源代码是否具有缺陷倾向性;

[0013] 设缺陷子树中有N个节点,则邻接矩阵A为N×N的矩阵,若两个节点间存在连接则在邻接矩阵中对应的元素为1,否则为0;设节点属性数量为D,则属性矩阵X是一个N×D的矩阵,节点的属性包括:节点是否发生修改,对应主题单词的一维向量中的各元素。

[0014] 本发明与现有技术相比,具有以下优势和积极效果:

[0015] (1)本发明提出一种基于抽象语法树结构特征学习的软件缺陷预测方法,利用软件的历史缺陷代码及其语义信息,使用机器学习的方法自动提取软件缺陷特征,利用深度学习的方法直接从结构化的软件代码中提取缺陷深度特征,解决了特征工程中度量元设计困难的问题,适合用于实际中面向开源软件的缺陷预测;

[0016] (2)本发明方法中使用深度学习得到的特征建立软件缺陷预测模型,对软件模块的缺陷倾向性进行预测,能够取得更好的缺陷识别效果,经实际验证用于实际软件工程项目中效果比较好。

附图说明

[0017] 图1是本发明的基于抽象语法树结构特征学习的软件缺陷预测方法的流程图。

具体实施方式

[0018] 为了便于本领域普通技术人员理解和实施本发明，下面结合附图和实施例对本发明作进一步的详细描述。

[0019] 本发明提出一种基于抽象语法树结构特征学习的软件缺陷预测方法，首先将源代码表示为抽象语法树，再利用修复信息和社团划分进行抽象语法树的裁剪，得到缺陷子树，并将修复描述和项目信息等背景信息综合到缺陷子树中进行缺陷预测，在预测时提出使用图卷积神经网络学习缺陷子树的有效表达，最后能获得比较好的缺陷识别效果，整体流程如图1所示，主要包括五个步骤，下面分别说明各步骤的实现。

[0020] 步骤一，采集开源软件缺陷信息，每条缺陷信息包括软件修复前、修复后两个版本的软件模块源代码，修复描述和项目基础信息，并以此构建软件缺陷数据库。

[0021] 假设开源软件托管库中开发人员每次标注的缺陷修复是有效的，修复前的模块为包含缺陷的软件代码，修复后的模块为不包含该缺陷的软件代码，每个缺陷的修复过程都会对应修复前和修复后两段源代码。开发人员在修复缺陷的同时，往往会通过Pull Request和Commit功能描述该缺陷的原因、种类和故障表现，此类描述可以称为修复描述，每条缺陷信息都包含文本类型的修复描述。每个缺陷都是存在于特定软件项目中的，每条缺陷信息中还包含所对应的软件项目的名称、主题、描述和语言等项目基础信息，项目基础信息同样以文本类型保存在软件缺陷数据库中。

[0022] 步骤二，根据步骤一获得的软件缺陷数据库，从中依次取出每个缺陷对应的软件模块的修复前和修复后的两部分源代码，以此分别生成两个抽象语法树，通过对比两个抽象语法树结构，得到抽象语法树中的差异节点，并将差异节点信息作为属性标记在两个抽象语法树的节点中。具体的，步骤二包含以下3个步骤：

[0023] 步骤201、将软件模块在修复前和修复后的源代码解析为抽象语法树。

[0024] 抽象语法树是软件源代码的抽象语法结构的树状表现，通过代码解析工具将上述软件缺陷数据库的源代码解析为抽象语法树，抽象语法树上的每个节点包含节点类型、代码内容和子节点集合等信息，修复描述信息和项目基础信息作为抽象语法树属性储存在抽象语法树中，此时软件缺陷数据库可表示为$D=((T_{11},T_{10}),(T_{21},T_{20}),...(T_{n1},T_{n0}))$，其中，n为缺陷信息总数，每条缺陷信息中包括修复前和修复后两个抽象语法树(T_{i1},T_{i0}) $(1\leqslant i\leqslant n)$，$T_{i1}$表示修复前的抽象语法树，$T_{i0}$表示修复后的抽象语法树；每个抽象语法树作为一个数据样本，则含有缺陷的数据样本数等于无缺陷的数据样本数。

[0025] 步骤202、抽象语法树差异分析。

[0026] 将每条缺陷信息的两个抽象语法树(T_{i1},T_{i0})的节点进行匹配，首先将抽象语法树的匹配问题看作从修改前的抽象语法树T_{i1}通过一系列的编辑操作转化为修改后的抽象语法树T_{i0}，这一系列的操作称为编辑脚本。抽象语法树的匹配过程是在两个抽象语法树的相似节点间建立映射关系的过程，其中一个抽象语法树上的节点只能添加到一个映射节点对中，匹配过程主要分为以下两个步骤：①采用贪婪算法自顶向下的寻找抽象语法树的同构子树，在各同构子树的节点间建立映射；②如果两个节点的后代中存在大量公共节点时，采

用自底向上的策略建立节点映射，当两个节点建立连接后，使用最优化算法再寻找后代中可能存在映射的节点。

[0027] 步骤203、标记修改节点。求解编辑脚本的过程伴随着树间节点的匹配，匹配成功的节点看作未发生修改的非缺陷代码节点；无法匹配的节点看作发生修改行为的缺陷代码节点，即发生差异节点；进一步的将是否发生修改作为属性储存在抽象语法树的节点中。

[0028] 步骤三，利用社团检测算法对每个缺陷的两个抽象语法树状结构进行社团划分，将所有包含差异节点的社团进行最小化连接，得到包含缺陷信息的缺陷子树。

[0029] 含有节点修改信息的抽象语法树往往包含大量缺陷无关节点和其他噪声节点，因此需要对抽象语法树进行剪枝。本发明方法通过Louvain社团检测算法结合节点修改信息对抽象语法树进行剪枝。

[0030] Louvain社团检测算法是一种基于模块度的社团检测算法，它能够有效的发现层次性的社团结构，得到整个社团网络模块度最大的结果，将抽象语法树看作有向无环图，对树状结构进行社团划分。对于抽象语法树，Louvain社团检测算法中的模块度Q计算如下：

[0031] $$Q=\frac{1}{2m}\sum_{i,j}[A_{i,j}-\frac{k_ik_j}{2m}\sigma(c_i,c_j)]$$

[0032] 其中，$A_{i,j}$为网络中节点i和节点j相连的边数，k_i为与节点i相连的边数，k_j为与节点j相连的边数，m为网络中边的总数，$\sigma(c_i,c_i)$函数表示若节点i与节点j在同一个社团中则返回1，否则返回0。c_i表示节点i所属的社团、c_j表示节点j所属的社团。

[0033] Louvain算法优化目标为整个社团模块度最大，它将每个节点初始化为一个社团，算法包括两个阶段：第一阶段尝试将单个节点加入能够使得模块度提升最大的邻近的节点所属社团；第二阶段将每个社团构造为一个超节点，迭代两个阶段直至社团划分结果不再变化。

[0034] 社团检测算法将抽象语法树划分为若干子树，表示为$T(r,V)=\Sigma T_i(r_i,V_i)$，T为抽象语法树，r为抽象语法树的根节点，V为抽象语法树的所有节点集合，T_i为第i个子树，r_i为子树T_i的根节点，V_i为子树T_i的所有节点集合，由此输出各个节点的子树归属。查找抽象语法树中的所有差异节点的子树归属，可以得到所有包含差异节点的子树集合。由于从子树集合中的若干子树可能具有不连通性，因此通过子树根节点间的最短路径将子树集合进行最小化连接，将得到子树作为缺陷子树，剪枝后的缺陷子树组成的数据库可以表示为：$D^*=((T_{11}^*,T_{10}^*),(T_{21}^*,T_{20}^*),\dots(T_{n1}^*,T_{n0}^*))$。$T_{n1}^*$表示剪枝后修复前的抽象语法树，$T_{n0}^*$表示剪枝后修复后的抽象语法树。

[0035] 也可以采用其他社团检测算法对抽象语法树进行子树划分，本发明实施例中选用了社团划分效果较优的Louvain算法。

[0036] 步骤四，提取项目基础信息、修复描述、以及步骤三中缺陷子树的类名、方法名、成员变量名等语义信息，利用主题建模技术获取缺陷子树的主题，进而利用文本的分布式表示方法将主题信息转化为向量数据作为缺陷子树的属性。具体的，步骤四包含以下3个子步骤。

[0037] 步骤401、文本信息预处理。

[0038] 每个缺陷子树对应其项目基础信息和修复描述构成了缺陷子树的背景信息；缺陷

子树中提取所在类的类名、成员变量名、方法名、方法返回值、方法形参名、方法形参类型及其他变量名，这些信息构成了缺陷子树的自身语义信息。缺陷子树的背景信息和自身信息都属于文本信息，使用自然语言处理的方法处理得到缺陷子树的信息语料库。具体的，步骤401文本信息预处理的处理方法包括：

[0039] a)分词。根据空格、标点符号和段落等分割方式，将文本信息划分为单词组；代码中的名称和类型往往是多个单词复合而成，根据常见的大写区分和下划线区分两种命名方式，将代码中名称和类型进一步的进行拆分，如将GetToken拆分为[Get,Token]，将get_request_token拆分为[get,request,token]。

[0040] b)停用词过滤。过滤英语语言中的停用词，如“the”、“and”等单词；过滤编程语言中的关键字，如“for”，“if，“return”等单词；过滤常用的无具体含义的编程相关词汇，如“main”，“arg”等。

[0041] c)提取词干。英文单词存在单数复数的变形，过去分词和进行时态的变形，在计算前应当使用提取词干的方式将一个单词的不同表达还原为一个单词对待，比如“stop”、“stopping”，“stopped”，“stops”应当合并为一个单词。

[0042] 步骤402、主题建模。

[0043] 经过步骤401处理后，每个缺陷子树将会对应一个语料库，语料库由若干单词组成，即包含了由项目基础信息和修复描述构成的背景信息以及自身的语义信息。主题建模需要从语料库中提取出最能表达各个主题的一些关键词，隐含狄利克雷分布(LDA)是一种常见的主题建模方法，其假设主题分布跟词分布由Dirichlet先验随机确定，然后通过Gibbs采样的方法估计主题分布和词分布，使用LDA提取缺陷子树语料库的主题以此构造缺陷子树的语义特征。考虑代码语义和语料库规模，取每个缺陷子树的主题数为1，LDA在该主题下取词分布结果，使用概率最高的5个单词代表该主题。

[0044] 步骤403、文本特征向量化。

[0045] 文本型特征无法直接作为模型输入进行学习，需要先将文本型特征转化为数值型的向量特征，常见的词向量化方法有独热编码和分布式表达两种，独热编码转换后的向量距离无实际意义，且容易引起维数灾难，本发明实施例使用google开源的word2vec方法对将文本型特征向量化。Word2vec是一种基于分布式假设的分布式表达方法，该方法假定出现在相同上下文中的词的意思应该相近，使用一层神经网络预测每个词的向量表示。Word2vec方法将缺陷子树的语料库每个主题单词映射为10维向量，每个缺陷子树的语义特征矩阵$S_{m\times h}$是一个5×10的矩阵，进一步的对语义矩阵压缩为一维向量，求矩阵每一列元素的均值作为一维向量对应列元素的值，压缩后的矩阵元素表示为$s_h^* = mean(s_{i,h})$，mean($s_{i,h}$)表示求取语义特征矩阵$S_{m\times h}$中第h列m个元素$s_{i,h}$(i＝1,2,...m)的均值，s_h^*表示一维向量的第h个元素，将压缩后的语义向量添加到缺陷子树节点的属性中。

[0046] 步骤五，建立基于图分类的卷积神经网络的软件缺陷预测模型，将步骤四中缺陷子树表示为邻接矩阵和属性矩阵作为模型的输入训练神经网络，识别待预测软件模块源代码是否具有缺陷倾向性。

[0047] 使用邻接矩阵和属性矩阵表示缺陷子树。邻接矩阵是树状结构的常见表达，对于包含N个节点的缺陷子树，邻接矩阵A可表示为N×N的0-1矩阵，若两个节点之间存在连接则

邻接矩阵对应元素为1，若两个节点之间无连接则该元素为0；若节点属性数为D，属性矩阵X是一个N×D的矩阵，矩阵X中的第i行元素记录缺陷子树的节点i的各特征属性x_i的数值。节点i的特征属性包括有步骤四计算的一维向量的h个元素，以及步骤二中是否发生修改的属性，节点是否发生修改用数字0和1来表征。

[0048] 图分类的卷积神经网络将卷积在非欧式结构上推广为节点聚合技术，通过不断地进行节点聚合自动的提取图中的结构特征，是一种端到端的学习方式，可以学习到更高层次的特征与模式，从l层聚合到l+1层的传播模型$H^{(l+1)}$可以表示为：

[0049] $H^{(l+1)}=f(H^{(l)},A)=\sigma(AH^{(l)}W^{(l)})$

[0050] 其中，$H^{(0)}=X$，$H^{(L)}=Z$，Z是第L层的输出，L是网络总层数，$l\in L$；σ是ReLU非线性激活函数，$W^{(l)}$是第l层的权值矩阵。在若干卷积层后使用池化(Pooling)技术获得用于图分类的图聚合结果，通过全连接层综合网络中学习到的所有信息，最后使用Softmax分类器输出缺陷倾向。

[0051] 本发明方法使用缺陷数据库中的所有缺陷子树作为图分类的卷积神经网络的训练数据，通过梯度下降算法迭代求解最小化损失函数，直至模型预测准确率不在提升，此时可以认为该模型已经具备了识别预测源代码缺陷的能力，预测时，将待预测源代码转化为树状结构，从该源代码中的语义信息和项目信息中提取特征，按照本发明步骤4先进行主题提取，再利用文本的分布式表示方法将主题单词转化为一维向量，作为节点属性，再通过训练好的模型就可以预测待预测源代码的缺陷倾向性。

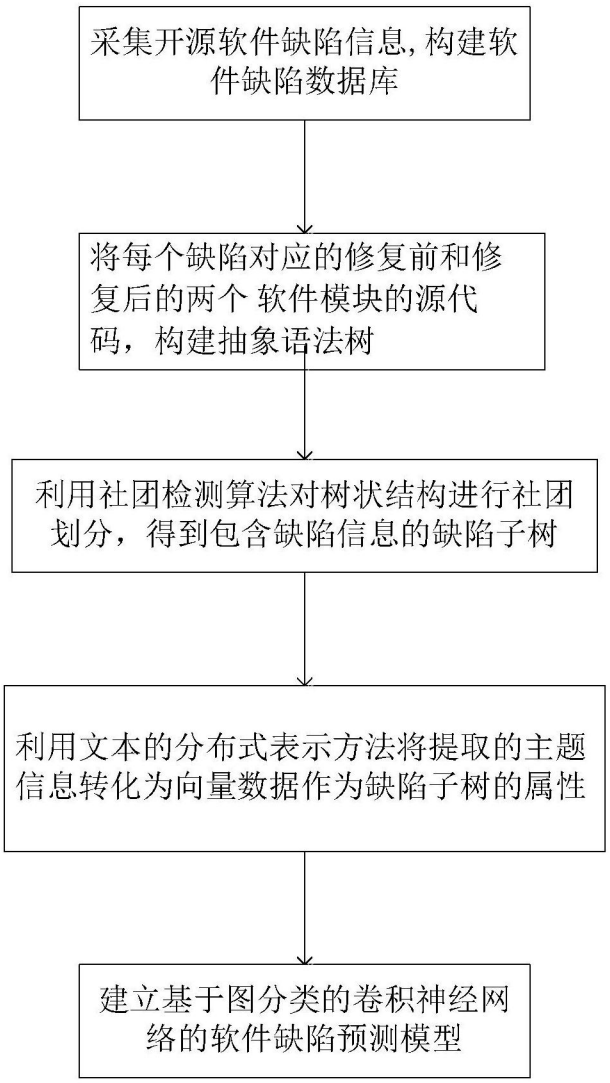

采集开源软件缺陷信息,构建软件缺陷数据库
将每个缺陷对应的修复前和修复后的两个 软件模块的源代码，构建抽象语法树
利用社团检测算法对树状结构进行社团划分，得到包含缺陷信息的缺陷子树
利用文本的分布式表示方法将提取的主题信息转化为向量数据作为缺陷子树的属性
建立基于图分类的卷积神经网络的软件缺陷预测模型

图1