第 36 卷 第 4 期 计 算 机 工 程 2010 年 2 月
Vol.36 No.4 Computer Engineering February 2010
·软件技术与数据库· 文章编号：1000—3428(2010)04—0045—02 文献标识码：A 中图分类号：N945

程序代码相似度度量的研究与实现

于海英
(内蒙古财经学院计算机信息管理学院，呼和浩特 010070)

摘 要：针对程序代码相似度的度量问题，提出一种属性计数和结构度量相结合的方法，通过统计程序源代码的操作符和操作数个数，产生 Halstead 长度、Halstead 词汇和 Halstead 容量 3 个程序的特征向量，利用向量夹角的余弦计算属性相似度，采用最长公共子序列算法获取结构相似度，从而衡量程序对间的相似程度。实验结果表明，该方法能够有效检测出学生作业中的相似程序代码。
关键词：属性计数；结构度量；程序代码相似度

Research and Implementation of Program Code Similarity Measurement

YU Hai-ying
(College of Computer Information Management, Inner Mongolia Finance and Economics College, hohhot 010070)

【Abstract】 Aiming at the problem of program code similarity measurement, a combined method of attribute counting and structure metrics is proposed. Attribute counting produces Halstead length, Halstead vocabulary and Halstead volume which constitute feature vector by counting the operator and operand of program source code, and attribute similarity can be calculated by using the cosine of vector included angle. The longest common subsequence algorithm is used to obtain structure similarity. The similar degree between two programs can be measured with the two similarities. Experimental results show the method can effectively detect similar programs of the students' homework.
【Key words】 attribute counting; structure metrics; program code similarity

1 概述

一种程序语言，对于同一逻辑的表达形式往往是多样的，还有可能一些人为了节省时间和力气，将别人的程序更改一下作为自己的。文献[1]将更改手段(在不影响结果的情况下)总结如下：(1)逐字拷贝；(2)更改注释语句；(3)更改空白区域和格式；(4)重新命名标识符；(5)改变代码块的顺序；(6)改变代码块中语句的顺序；(7)改变表达式中操作符和操作数的顺序；(8)更改数据类型；(9)增加冗余的语句和变量；(10)用等价的控制结构替换原有控制结构。这些更改都是表面的、少量的，而程序中内含的属性和结构没有改变。因此，这样的程序具有内在的相似性，且这种相似性是可以度量的。本文主要探讨这种相似性的度量方法。

2 相关研究

程序代码相似度是指利用一定的检测方法度量程序代码间的相似程度，主要应用在程序的复制检测上，因此，程序代码相似度度量技术的发展是随着程序复制检测技术的发展而发展的。除了逐字拷贝的情况，使用直接比较 2 个程序文本文件字符串的方法来度量相似度的策略效率非常低。在国外，最早在 20 世纪 70 年代初就有学者研究度量程序代码相似度的技术和软件。Ottenstein 在 1976 年首次提出基于属性计数法度量相似度的方法，也有学者沿用了属性计数的方法。Plague、YAP 系列工具、SIM[2]、JPlag 使用结构度量的方法来度量程序的相似度。Verco 等人指出，单纯的属性计数法抛弃了太多的程序结构信息，增加向量维数并不能改善错误率。改进属性计数法的措施就是加入程序的结构信息，结合结构度量来度量相似度。在国内，1988 年中国人民警官大学的张文典、任冬伟研制了一个 PASCAL 程序抄袭判定系统，利用属性计数的方法能够度量 PASCAL 程序的相似度。以后很少有人在该领域进行研究。

3 程序代码相似度度量

以 C 语言为度量语言，测试数据选用某班学生上传到 FTP 上的同一批次的数据结构堆栈操作作业(用 C 语言实现)文本文件。属性计数时用到 C 语言的关键字和操作符号字典，需要预先建立。实现时分为 3 步：

第 1 步 预处理：删除注释语句；将语句中出现的如“___”这样的系统不能识别的绘图符号，用系统能够识别的符号(如“*”号)代替，避免出现运行错误。

第 2 步 属性计数：将每个程序读进内存并按程序语句自顶向下和代码行从左向右的顺序进行分解，得到构成程序的标识符序列，统计各属性的值，并将标识符序列依次存入数据库中，为下一阶段做好准备。

第 3 步 结构度量：比较构成 2 个程序的标识符序列，获取最长公共标识符子序列的长度，如果需要也可以获取最长公共标识符子序列。

3.1 程序属性相似度的度量

文献[3]认为任何计算机程序能够被统计或度量的属性包括：η_1(程序中唯一的操作符数)；η_2(程序中唯一的操作数数)；N_1(程序中总的操作符数)；N_2(程序中总的操作数数)。

基金项目：内蒙古财经学院科研基金资助重点项目(KY0642)；内蒙古自治区高等学校科学研究基金资助项目(NJ09125)
作者简介：于海英(1976－)，女，讲师、硕士，主研方向：数据挖掘
收稿日期：2009-11-10 **E-mail：**yuhaiying@163.com

从基本的度量值很容易得到程序的词汇数：$\eta=\eta_1+\eta_2$；程序的实施长度：$N=N_1+N_2$；程序的容量：$N\text{lb}\eta$。这成为程序可以度量的理论基础。

3.1.1 Halstead 属性

本研究采用 M.Halstead 对程序属性的分析方法，根据程序标识符的类型和它们出现的频度来标识程序。主要统计 2 种类型的标识符：操作符和操作数。对于不可执行代码行，如注释、空行和空格，将其忽略。操作符标识符包括源程序设计语言的关键字、操作符号和标准函数名。操作数是由程序设计者自定义的标识符。这些数值用来产生：Halstead 长度(N)，Halstead 词汇(η)，Halstead 容量($N\text{lb}\eta$)。由于属性统计的不止一个方面，因此每个程序的属性构成一个特征向量 $\boldsymbol{v}_i$，进而可以将属性相似度的度量问题转化为特征向量的距离计算问题。

3.1.2 属性相似度计算

2 个程序的特征向量获取后，可以通过向量夹角的余弦来计算相似度。为了避免在向量空间中过于强调向量的各个坐标的绝对值，将向量进行归一化处理。假设 $\boldsymbol{v}_i$ 为归一化后的向量，由于归一化后向量夹角的余弦正好是 2 个向量之间的点乘，则相似度计算公式为

$$sim(\boldsymbol{v}_i,\boldsymbol{v}_j)=\boldsymbol{v}_i\cdot\boldsymbol{v}_j=\sum_{k=1}^{m}V_{ki}\times V_{kj}\quad 0\le sim(\boldsymbol{v}_i,\boldsymbol{v}_j)\le 1 \tag{1}$$

$sim(\boldsymbol{v}_i,\boldsymbol{v}_j)$ 越接近 1，说明作比较的 2 个程序 $\boldsymbol{v}_i$ 与 $\boldsymbol{v}_j$ 相似越密切；若等于 1，则说明 2 个程序 $\boldsymbol{v}_i$ 与 $\boldsymbol{v}_j$ 为同一个程序或 2 个程序完全相同，或者是在没有改变程序结构和标识符个数的情况下拷贝生成另一个程序；反之亦然，但由于 C 语言程序的总体结构相同(使用同样的操作符号和关键字)，$sim(\boldsymbol{v}_i,\boldsymbol{v}_j)$ =0 的情况很难达到。

3.2 结构相似度度量

最长公共子序列[4](Longest Common Subsequence, LCS)是将 2 个给定字符串分别删去零个或多个字符，但不改变剩余字符的顺序后得到的长度最长的相同字符序列。在结构度量时，要查找 2 个标识符序列的最长公共子序列，称之为最长公共标识符子序列，它的长度称为最优值。

通过属性度量获取程序对的标识符序列后，根据应用动态规划思想的最长公共子序列算法计算最优值，进而得到最长公共标识符子序列。要计算标识符表 P_i 和 Q_j 的最优值，用初值为 0 的 $c(m,n)$数组，其中，m 和 n 分别为 P_i 和 Q_j 的长度。记录整个计算过程中的最优值矩阵，当 i=0 或 j=0 时，空序列是 P_i 和 Q_j 的最长公共标识符子序列，故 $c(i,j)$=0。其他情况下，可自底向上建立递推关系如下：

$$c(i,j)=\begin{cases}0 & \text{当}i=0\text{或}j=0\text{时}\\ c(i+1,j+1)+1 & \text{当}i,j>0\text{且}p_i=q_j\text{时}\\ \max(c(i+1,j),c(i,j+1)) & \text{当}i,j>0\text{且}p_i\ne q_j\text{时}\end{cases} \tag{2}$$

由递推关系结合动态规划算法很容易写出自底向上计算最优值 $c(i,j)$的算法 LCS-LENGTH 及自顶向下获取最长公共标识符子序列算法 LCS，2 个程序的最长公共标识符子序列存入数据表中，以供查看。程序对的最优值 $c(0,0)$获取后，通过计算 Strus1=$c(0,0)/m$ 和 Strus2=$c(0,0)/n$，获取结构相似度值。结构相似度取值范围为：0%~100%；Strus1 和 Strus2 越接近 100%，说明 2 个程序代码结构相似越密切；都等于 100%，说明作比较的 2 个程序为同一个程序或 2 个程序完全相同；反之亦然，但由于 C 语言程序的总体结构相同(使用同样的操作符号和关键字)，都等于 0 的情况很难达到。程序对的属性和结构相似度获取后，可以将其以降序排列方式写入文本文件进行保存和查看。

3.3 程序代码相似度的阈值

阈值的大小决定程序间相似到一种什么程度才称之为相似程序。如阈值偏大，则找到的相似程序就少；若阈值偏小，则找到的相似程序就过多。对于不同批次的作业，可以得到不同范围和分布的相似度值，但是并没有一个确定的临界值(阈值)，超过该阈值可以准确判断 2 个程序是相似的[1]。由此可以看出，阈值必须选得恰当。因此，每次只能根据手边的数据和实际情况，利用统计规律，反复比较、反复修改来确定阈值。

4 实验测试与结果分析

4.1 简单程序测试与分析

对于 SIM 中的 2 个实验程序 TEST1 和 TEST2(TEST1 是原始程序，TEST2 是由 TEST1 经过删除注释、改变所有变量和函数名称、尽可能颠倒相邻语句的顺序、改变函数的顺序等操作得到的)，本研究报告 Halstead 相似度为：0.999 997 7；结构相似度为：46.89%和 44.39%。虽然结构相似度不高，但 Halstead 相似度偏高，说明 2 个程序是相似的。由于 TEST2 在 TEST1 的基础上做了若干结构的改变，因此结构相似度值不高，这是可以理解的。SIM 对这 2 个程序进行测试时，最后获取的相似度值为 0.4(相似度取值范围为 0.0~1.0，值越大程序越相似)。SIM 认为这 2 个程序有相似的可能，值得做进一步考察。

4.2 批量程序测试与分析

利用本研究对小批量的 39 个实验程序进行测试，其中最小为 272 Byte，最大为 2 532 Byte，共有 741 个程序对。运行后根据相似度值用 Minitab15 统计软件生成分析如图 1、图 2 所示，其中，图 1 的属性相似度均值为 0.999 8，标准差为 0.000 474 5；图 2 中 2 个变量的均值与标准差分别为 0.491 4, 0.159 7; 0.541 2, 0.140 4。

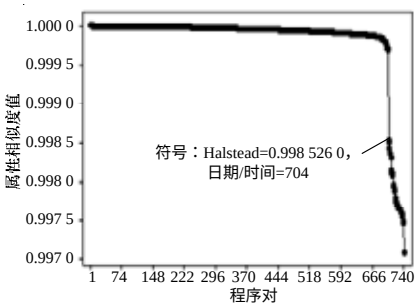

图 1 Halstead 属性相似度时间序列

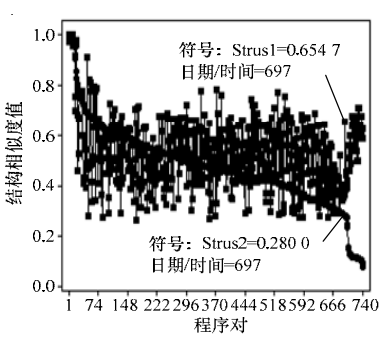

图 2 结构相似度时间序列

(下转第 49 页)