プログラム間の類似性の定量化手法（２）

Method of Similarity Quantification between Program Codes (2)

上村　康輔[*1]，若林 茂[*2]
Kosuke KAMIMURA[*1], Shigeru WAKABAYASHI[*2]
[*1] 神戸市立工業高等専門学校専攻科　電気電子工学専攻
[*1]Department of Electrical and Electronic Engineering, Kobe City College of Technology
[*2] 神戸市立工業高等専門学校　電子工学科
[*2] Department of Electronic Engineering, Kobe City College of Technology
Email: r108310@g.kobe-kosen.ac.jp

あらまし：プログラミング教育の現場では，教員は学生から提出されるプログラムの評価を行う必要がある．これは教員の負担となるので，プログラム間の類似性を機械的に評価できれば教員が採点を行う際の負担を軽減できると考えた．本研究では，類似性を評価する手法として構文木を用いたカーネル法を使用していた．本稿では，カーネル法の補助を目的としてプログラムの構成要素数から類似度を定量化する手法を新たに追加する．
キーワード：プログラミング教育，プログラム間の類似性，カーネル法，クラスタリング

1. はじめに

プログラミング教育の現場では，教員は学生から提出される大量のプログラムを目視により妥当性を判断し，それぞれのプログラムに適した評価を行う必要がある．授業で出題される課題は簡単なアルゴリズムで解ける問題が多く，提出されるプログラムはほとんど差異の無いものとなる．教員はこれらのプログラムを何度も評価することになるため，評価を行うことが負担となる．また，教員が間違えた評価をする可能性も存在する．そこで，プログラム間の類似性を機械的に評価できれば，教員の負担を軽減できると考えた．以前の研究では，入出力の比較や制御構造の比較などの比較法を感度が高いと考えられるものから順に比較する段階的感度比較法を採用していた[(1)]．しかし，この手法は順序尺度や間隔尺度に問題がある．

本稿では，本研究で使用していたカーネル法についての説明と，カーネル法の問題点を補うための新たな手法について述べる．

2. カーネルを用いた類似性の定量化[(2)]

2.1 カーネル法とは

本研究では，類似性を求める操作としてカーネル法を用いていた．カーネル法とは，データ構造間の内積に相当するスカラ値を定義することによって，内積空間上の種々のアルゴリズムを一般のデータ構造へ適用できるようにする方法である．この内積を求める関数をカーネルと呼ぶ．プログラムはラベル付き木として表現できるので，木に関する既知のカーネルを使用できる．これにより，プログラム間の類似度を数学的に解析できる．

2.2 カーネル法の問題点

カーネル法では，類似度の計算を部分木の内容の比較によって求めており値自体は意味を持たない．つまり，出力された値だけではプログラムのどの部分が異なるのかといった違いの詳細を知ることができない．さらに，この手法で求められる類似度は，式の順序や変数の使い方などのコーディング手法の違いでは大きく変化するが，プログラムの計算量に関する違いはコーディング手法の違いよりも小さな変化となる．これより，プログラムの計算量に関する違いよりもプログラムの構造の違いの方が重要視されてしまう．

3. 新たな手法

本研究ではカーネル法の問題点を補うことを目的として，プログラムの構成要素の数から類似度を定量化する方法を提案した．比較対象の要素として

- 変数の数
- 関数を呼び出した回数
- 演算子の数
- 選択文の数
- 反復文の数

の５つを定義し，２つのプログラム間のユークリッド距離を類似度とした．それぞれの構成要素についての詳細を以下に示す．

① 変数の数
これは変数が宣言される数をカウントした．型ごとの区別は無く，配列も 1 つの変数としてカウントする．

② 関数を呼び出した回数
これは関数，手続きを呼び出した回数をカウントした．関数の中には，平方根を求める sqrt 関数といった数学関数なども含まれる．

③ 演算子の数
これはソースコード中の’ +’, ’ -’, ’ *’, ’ /’ の数をカウントした．ただし，選択文，反復文の条件として使われる演算子，配列の添え

字として使われる演算子はカウントせず，算術代入文の右辺で使われる演算子のみをカウントした．

④ 選択文の数
これは if 文，case 文といった選択文の数をカウントした．case 文については，それぞれの条件を 1 つの選択文としてカウントし，if 文についても else if は 1 つの選択文としてカウントした．ただし， else についてはカウントしない．

⑤ 反復文の数
これは for 文，while 文といった反復文の数をカウントした．for 文については，ソースコードには書かれていないが繰り返すごとに変数の値も増えるため，同時に演算子の数も 1 増やしている．

また，得られたユークリッド距離に対してクラスタリングを行った．

4. プログラムの解析

神戸高専のプログラミング I の授業で提出された，1000 までの素数を判定するプログラムの解析結果を図 1 に示す．

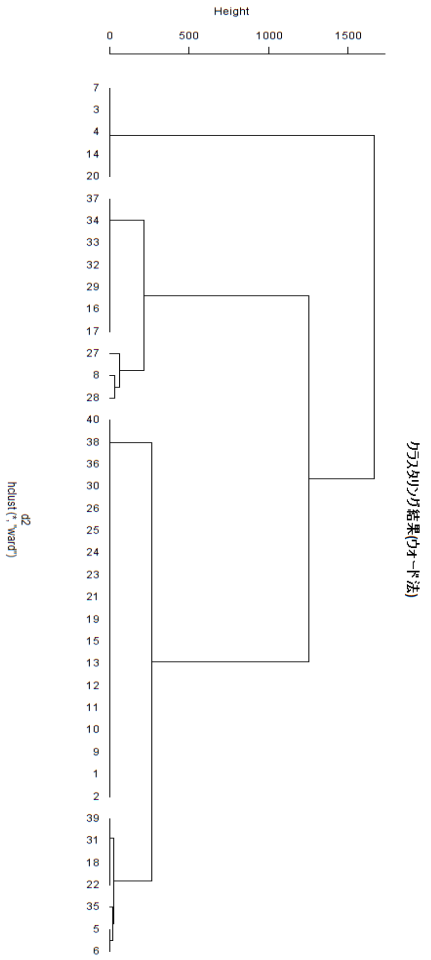

図 1：解析結果

図 1 のグラフはデンドログラム（樹形図）というグラフで，各個体がどのようにクラスタとしてまとめられていくかを表したものであり，縦に並んでいる数字は出席番号を示している．この図から提出されたプログラムは表 1 のように分類される．

表 1 ：分類結果

グループ	出席番号
グループ A	3,4,7,14,20
グループ B	16,17,29,32,33,34,37
グループ C	8,27,28
グループ D	1,2,9,10,11,12,13,15,19,21, 23,24,25,26,30,36,38,40
グループ E	5,6,18,22,31,35,39

表 1 と実際のプログラムを照らし合わせてみると，グループ A は未提出のグループであり，グループ C は題意を満たしていないグループであった．それ以外のグループは題意を満たしており，特にグループ B では素数の判定を$\sqrt{n}$まで，グループ D では n−1 まで繰り返しを行っていた．グループ E に関しては，繰り返し回数はグループ D と同じだが，if 文の数が多いなど独自の書き方をしているプログラムが多く存在した．このことから，この手法によって求められた類似度はプログラムの計算量に関する違いが大きく影響していることが考えられる．

5. まとめ

カーネル法の問題点を補うための新たな手法としてプログラムの構成要素の数から類似度を求めた．また，求めた類似度に対してクラスタリングを行うことでプログラムの計算量が似ているいくつかのグループに分類することができた．今後の研究では，定義の細分化といった手法の改善や，カーネル法の結果も含め教員がより確認しやすくなるためのデータの可視化を行っていく必要がある．

参考文献

(1) 井上晴喜，若林茂：“プログラム間の類似性に関する研究”, 教育システム情報学会第 31 回全国大会講演論文集，教育システム情報学会，pp.401-402（2006）
(2) 小田悠介，上村康輔，若林茂：“プログラム間の類似性の定量化手法”，JSiSE，第 37 回全国大会講演論文集，B4-2（2012）