3ZA-09

プログラミング初学者のソースコード記述特徴抽出方法の検討

浅田 理芳† 山崎 和也† Martin J. Dürst† 戸辺 義人†
青山学院大学理工学部情報テクノロジー学科†

1. はじめに

プログラミングの教育現場では，課題として学生にソースコードを提出させることが一般的である．教師側は受講する学生のコーディングの癖を把握することは，課題の難易度や授業の組み立てを考える際に役立つ．また，学生間での課題のコピーは，正しい成績評価や技術修得の妨げとなるため，提出物の類似性を調べる必要がある．そこで本研究では，ソースコードの記述方法の特徴から，コーディングした学生を類別する．ソースコードの記述方法は人によって差異があり，個人の癖が表れる．こうした癖を抽出し，C 言語ソースコードの分類手法を提案する．

2. 関連研究

ソースコードから表層的な書き方の特徴であるコーディングスタイルを特徴量として抽出し，盗用関係を検証する研究がある [1]．これらの研究では，「インデント，ファイルサイズ」「コメント」の量，「括弧」「演算子」の前後にある空白，「変数や関数の名前」という 5 つに分類される特徴の抽出を行っている．いずれも割合や量で表される特徴であり，ソースコードの中身ではなく，見た目で分かる特徴を使用している．これに対し，本研究では，制御構造などのソースコードの中身に着目し，有無で表すことができる特徴によって分類を行った．

3. 抽出する特徴

特徴は 11 種類として，すでに終了したプログラミングの授業で提出された C 言語ソースコードより抽出を行う．それぞれの特徴は，0，1 の 2 つに分類するものとする．ただし，変数，配列，構造体に関してのみ 0，1，2 の 3 つに分類する．特徴の有無で分類するものは，有る場合を 1，無い場合を 0 とする．以下，j 番目の課題の u 種類目の特徴をf_{ju}として示す．

3.1. 繰り返し処理

while 文を用いた繰り返し処理の記述方式の分類として，以下の 3 つの手法を定義し，それぞれの記述方式の有無を特徴とする．

f_{j1}：無限ループを作り，処理中に条件を満たしたとき，break 文で脱出する

f_{j2}：無限ループを作り，ユーザからの入力によって終了する

f_{i3}：while 文にあらかじめ終了条件を設定する

3.2. コメント (f_{j4})

コメントの有無を特徴とする．

3.3. 関数 (f_{j5})

プログラム中の関数の有無を特徴とする．ここでの関数とは，標準ライブラリ関数ではなく，ユーザ定義関数を指す．

3.4. ソート方法 (f_{j6})

データの並び替えの手法を特徴とし，その手法として，以下の 2 つを定義する．

0：main 関数内にソート方法を記述する．

1：関数を定義し，ソート方法を記述する．

3.5. 変数，配列，構造体 (f_{j7})

0：基本的な変数のみを使用している．

1：配列も使用している．

2：構造体も使用している．

3.6. 変数の宣言 (f_{j8})

変数宣言の記述方式の違いを特徴とし，以下の 2 つの記述方式を定義する．

0：同じ種類の型であれば，複数の変数を 1 行にまとめて宣言する．

1：同じ種類の型であっても，1 つまたは 2 つの変数ごとに分けて宣言する．

3.7. 条件分岐 (f_{j9})

複数の条件によって分岐を行う場合の条件分岐の記述方式の違いを特徴とする．以下の 2 つの記述方式を定義する．

0：if 文を用いて分岐を行う．

1：switch 文も用いて分岐を行う．

3.8. 結果出力の仕方 (f_{j10})

演算結果を printf 関数を用いて出力するときの記述方式の違いを特徴とする．

0：あらかじめ演算を行い，演算結果を変数に格納しておき，変数の中身を printf 関数によって出力する．

1：printf の引数内に計算式を記述し，演算結果

Study on Feature Extraction Method of Source Code Written by Novice Programmers

† Rika ASADA, Kazuya YAMAZAKI, Martin J. DÜRST, Yoshito TOBE/ Aoyama Gakuin University

Copyright ©2016 Information Processing Society of Japan.
All Rights Reserved.

を出力する．

3.9. 提出状況 (f_{j10})

全データの中には，提出数の少ない課題や学生が存在する．提出を1，未提出を0として11種類目の特徴に追加した．

4. 分類

4.1. 特徴ベクトルの作成

未提出の課題の特徴は，すべて0として処理すると，未提出の学生が多い課題では分類が難しくなる．したがって，課題と学生の二つの視点において使用するデータの絞込みを行った．絞り込んだデータから，学習に使用する特徴ベクトルを作成する． i番目の学生のj番目の課題の特徴ベクトル$\boldsymbol{f}_j^i$は以下のように表せる．

$$\boldsymbol{f}_j^i = (f_{j1}, f_{j2}, f_{j3}, \cdots, f_{j11})$$

この特徴ベクトル$\boldsymbol{f}_j^i$を用いて，学習に使用する特徴ベクトル$\boldsymbol{F}$を作成する．以下，手順を述べる．

① 学習に使用する課題をm個選択する．
② 特徴毎にm個の平均値を計算し，特徴ベクトルを作成する．

課題の絞り込み方と学習に使用する課題数mを変化させ，そのときに用いた組み合わせをパターンαとする．よって，i番目の学生のパターンαにおける特徴ベクトル$\boldsymbol{F}$の式は以下のように表せる．

$$\boldsymbol{F}_\alpha^i = (F_{\alpha1}, F_{\alpha2}, F_{\alpha3}, \cdots F_{\alpha11})$$

4.2. クラスタリング

作成した特徴ベクトルに対し，X-means 法を用いてクラスタリングを行う．ここで作成されたクラスタ数をKとする．Kの最小値と最大値を変化させ，場合分けをした．

5. 評価

5.1. 精度分析

特徴ベクトル$\boldsymbol{F}$に用いていない課題(以下，テスト課題)と，すべてのクラスタの中心点とのユークリッド距離を算出し，その値の最小となるクラスタにi番目の学生が属すると推定する．

5.2. 実験

今回は全データの中から，以下の条件におけるデータを使用した．

- 課題の提出率：0.6以上
- 課題における特徴の分散が有る：6つ以上
- 学生の提出率：0.8以上

課題数は18，学生数は58人である．その中で学習における特徴ベクトルFに使用する課題とその数mを変化させる．また，クラスタ数の最小値も5～7の間で変化させ，検証する．

5.3. 検証

検証の指標として，正確性を示す適合率の平均値，網羅性を示す再現率の平均値，F尺度を用いた．適合率と再現率は互いにトレードオフの関係であるため，F尺度がより高いものほど良い予測結果が得られていると考える．

5.4. 結果

特徴ベクトルⅠ，Ⅱは m=3，ⅢはⅠとⅡの課題を合わせてm=6の場合で検証を行った．表1にクラスタ数の最小値を5として，17番目のテスト課題を推定した場合の分類結果を示す．A,…,Eは特徴ベクトルを用いたときに分類されたクラスタ，$\hat{A},\cdots,\hat{E}$はユークリッド距離を用いて推定されたクラスタである．表2にそれらの平均適合率，平均再現率，F尺度を示す．

それぞれの比較を行うと，表2では平均適合率，平均再現率，F尺度はⅢが高いことがわかる．

表1 特徴ベクトルⅢの分類結果

	$\hat{A}$	$\hat{B}$	$\hat{C}$	$\hat{D}$	$\hat{E}$
A	5	0	0	0	0
B	1	5	6	2	3
C	3	4	7	3	3
D	0	0	0	4	0
E	2	3	1	4	1

表2 平均精度，平均再現率，F尺度

No.	平均適合率	平均再現率	F尺度
Ⅰ	0.35	0.39	0.37
Ⅱ	0.29	0.36	0.32
Ⅲ	0.36	0.55	0.44

6. 考察

正解率を算出すると，すべての場合において0.4前後ではあるものの，特徴ベクトルⅢでは再現率が1となるクラスタA,Dが現れた．現時点では本手法は，まだ期待できるF尺度となっていないため，今後さらに分類に必要な特徴を検討する．

7. むすび

本研究では，制御構造などのソースコードの中身の特徴量を癖として分類し，プログラマを推定するための予測モデルを提案し，検証した．

参考文献

1) 福本 大，北川 裕基,玉田 春昭：コーディングスタイルを基にした初心者プログラマ向けの著者解析： 情報処理学会研究報告，ソフトウェア工学（SE）， Vol.2015-SE-187，No.16，(2015).

Copyright ©2016 Information Processing Society of Japan.
All Rights Reserved.