1D4-2

前後方記述スタイルに基づいた 授業課題ソースコードのクラスタリング

Clustering In-class Source Codes Based on Forward-Backward Coding Style

大野　麻子[*1]　村尾　元[*2]
Asako Ohno　Hajime Murao

[*1]四條畷学園短期大学ライフデザイン総合学科
Department of Life Design, Shijonawate Gakuen Junior College

[*2]神戸大学大学院国際文化学研究科
Graduate School of Intercultural Studies, Kobe University

In this paper, we extracted a programmer's coding style from a number of source codes that had been produced by the programmer and represented as parameters of a set of stochastic models called "Forward-Backward Coding Models". We classified the programmers according to their coding styles by inputting feature vectors consisted of parameters of each of the models to Self-Organizing Map. The programmers placed closer on the map had similarities in their way of writing source codes.

1. はじめに

本研究では，複数のソースコードから作成者固有の書き方の特徴である「記述スタイル」を抽出し，隠れマルコフモデル (HMM) をベースとした複数個の確率モデル群に学習させることで，ソースコード作成者の特徴を定量表現している [Ohno 10]. 本稿では，これらのモデルのパラメータより生成した特徴ベクトルを自己組織化マップ (SOM) に入力し，記述スタイルのクラスタリングを行う．つまり，ソースコードを用いて間接的にその作成者の分類を行う．

自己組織化マップの出力結果として，2 次元マップ上に記述スタイルに基づく作成者の類似関係がマッピングされる．このような情報は，プログラミング授業支援を目的とした様々な用途に利用可能であると考えられる．例えば，教員の作成した手本も含め，授業中に学生の作成したソースコードを定期的に回収し，SOM によるクラスタリングを行うことで，学生の記述スタイルの変遷や個性が確立されていく過程を観測する試みや，成績情報と組み合わせることで，プログラミングが得意な学生と不得意な学生の記述スタイルにおける何らかの特徴を検出する試みなどが考えられる．また，教育現場において解決が必要とされる問題の一つである「授業課題ソースコード盗用問題」においては，既存手法の多くがソースコードの内容に基づいた類似性尺度を用いて盗用発見を行っているため，偶然の一致を盗用と誤判定してしまう可能性が危惧されており [Ohno 10]，単一の類似性尺度では盗用が成功してしまう可能性が高いが，複数の異なる類似性尺度を併用することにより，盗用発見の精度が向上する可能性も示唆されている [Ueda 08]. 本手法の類似性尺度である記述スタイルは，既存研究ではノイズとして除去されていた表面的な特徴に着目したものであり，既存の手法と併用することにより，様々なメリットが期待される．例えば，あるクラスにおいて授業中に作成されたソースコードと最終課題として提出されたソースコードの記述スタイルの分類結果を比較し，分類結果に大きな差異があれば，該当する学生のみについてその原因が盗用であるか否かを既存の盗用発見手法により調査するという用法は，教員の精神的・肉体的負担を軽減させることにつながる可能性がある．

連絡先: 大野麻子，四條畷学園短期大学ライフデザイン総合学科，〒 574-0011 大阪府大東市北条 4-10-25, asako.ohno@mulabo.org

2. 提案手法における定義

2.1 記述スタイル

本研究ではソースコード作成者がソースコードを記述する際に，意識的にまたは無意識のうちに付与される作成者固有の表記上の特徴を「記述スタイル」と呼ぶ．本研究ではソースコードから記述スタイル規則により切り出されたトークン部分列を記号系列，ソースコード作成者を記号系列の発生源 A_α とすると，A_α により生成される記号系列は常に A_α 固有の確率変数系列により表すことができ，この確率変数系列にはマルコフ性が成り立つとしている．つまり，記述スタイルはソースコードの内容に依らず同一の発生源から発生する記号系列の取りうる値を決定するマルコフ過程として定義される．

2.2 前後方記述スタイルモデル

作成者 A_α の記述スタイルを定量的に表現するのが，HMM をもとに考案された「前後方記述スタイルモデル (Forward-Backward Coding Model)」M_α である．M_α は,「前方記述スタイルモデル (Forward Coding Model)」M_1^α および「後方記述スタイルモデル (Backward Coding Model)」M_2^α により構成される．予め定義された "{" のような 14 種類の「基準トークン」$n_b (1 \leq n_b \leq 14)$ の前後における記述スタイルを，基準トークンの前方の記述スタイルを表す M_1^α と後方を表す M_2^α により表現する．具体的には，基準トークンの前方および後方における,「着目トークン」と呼ばれる 1〜4 文字分の空白，タブ，改行の並びを，それぞれ前方および後方の記述スタイルを内包する記号系列としてソースコードから全て切り出し，このような記号系列を発生させるように M_α 中の対応するモデルのパラメータを更新することで記述スタイルを表現する．

$$M_p^\alpha = \left\{ M_{p,1}^\alpha, \cdots, M_{p,n_b}^\alpha, \cdots, M_{p,14}^\alpha \right\}, p \in \{1,2\} \qquad (1)$$

$p \in \{1,2\}$ は基準トークンの前方または後方であるという位置情報を示す．M_p^α はそれぞれ，14 種類の基準トークンに対応した 14 個の記述スタイルモデルにより構成される．従って M_α は全部で 28 個のモデルにより構成される．

M_{p,n_b}^α はいずれも 3 個の状態を持ち，状態間の遷移確率，各状態における観測確率，各状態が開始状態か否かを表す初期確率，各状態から観測可能な記号の集合により構成される．

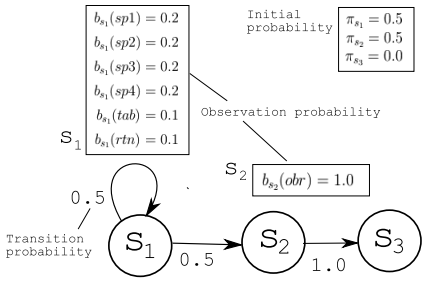

図 1: 前方記述スタイルモデルの例

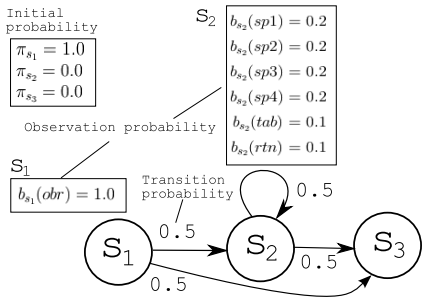

図 2: 後方記述スタイルモデルの例

図 1 および図 2 は基準トークンの 1 つ，“{”(obr) に関する記述スタイルを表すモデルの例である．図 1 に示す前方記述スタイルモデルは，着目トークンである 1〜4 文字分の空白 (sp1〜sp4)，タブ (tab)，改行 (rtn) がそれぞれ観測可能な状態 s_1 と基準トークンが観測可能な s_2，そして終了状態 s_3 により構成される．着目トークンが基準トークンの直前に表れない記述スタイルも存在するため，s_1，s_2 の状態の初期確率はいずれも 0.5 となる．次に，図 2 に示す後方記述スタイルモデルは，基準トークンが観測可能な状態 s_1 と着目トークンが観測可能な s_2，そして終了状態 s_3 により構成される．記述スタイルは必ず 1 個の基準トークンを含むため，常に s_1 の初期確率は 1.0，s_2 の初期確率は 0.0 である．

3. SOM によるクラスタリング

SOM は多次元のデータの分析器として幅広く利用されている競合学習型ニューラルネットワークの一種である [Kohonen 96]．本稿では，前後方記述スタイルモデルにより定量化された複数の作成者の記述スタイルを SOM により分類する．

まず，次の手順により入力データを生成する．12 名の被験者にそれぞれ 20 個ずつの Java プログラミング課題を与え，盗用の無いソースコードを作成させる．次に，同一人物により作成された 20 個のソースコードの記述スタイルをモデルに学習させ，12 名の記述スタイルを表す 12 セットの前後方記述スタイルモデル群を用意する．記述スタイルは，前後方記述スタイルモデル群を構成する各モデルのパラメータにより表される．本稿では，これらのパラメータを用いて記述スタイルの分類を行う．図 1 の各状態における初期確率，観測確率，遷移確率を並べ，続いて図 2 についても同様に行う．これを 14 種類全ての前後方記述スタイルモデルについて行い，448 次元の要素を持つ特徴ベクトルを形成する．これを 12 名の被験者全てについて作成し，SOM に入力した結果が図 3 である．実験には

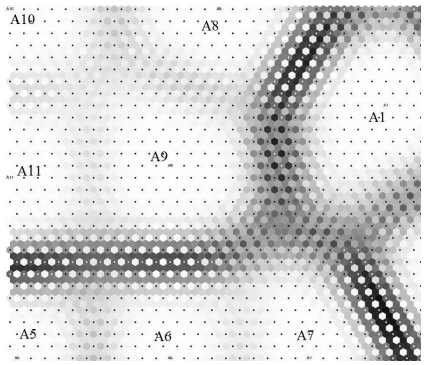

図 3: SOM によるクラスタリング結果

`som_pak frontend`[Ohkita 08] を使用した．ノード数は 40 × 30 個，初期学習係数は 0.2 とし，10000 回の学習を行った．

マップ上の英数字は作成者を表す．隣接して配置された作成者のソースコード間にはいくつかの記述上の類似点が確認された．しかし，作成者 $A1$ と $A5$ のように離れた場所にある作成者のソースコード間にも類似が見られた．これは本実験で用いた基本 SOM の特性によるものである可能性もある．

4. おわりに

本稿では，ソースコードから抽出した記述スタイルを定量化し，自己組織化マップにより分類するという試みについて報告した．このような情報は，記述スタイルと習熟度の関連や盗用発見など，プログラミング授業支援に有益な知識として活用されることが期待できる．SOM の出力結果からは，作成者の記述スタイルの類似を一覧することができ，目視でソースコードの記述を確認したところ，マップ上で近傍に配置された作成者のソースコード間には表記上の類似性が多く確認された．今後の課題として，更に詳細なレベルでの記述スタイルの分類を実際の授業課題ソースコードを用いて行うことなどがあげられる．

謝辞

本研究は科研費（課題番号：21800085）の助成を受けて行われた．ここに謝意を表す．

参考文献

[Ohno 10] 大野 麻子, 村尾 元: 前後方記述スタイルモデルによる授業課題ソースコード作成者特徴の抽出, 第 37 回知能システムシンポジウム資料, pp.99–104, (2010).

[Ueda 08] 上田 広明, 安間 文彦ほか: 複数の類似度計算指標を統合したソースコード剽窃検出システム, 人工知能学会先進的学習科学と工学研究会資料, vol.46, pp.55–60, (2008).

[Kohonen 96] T. コホネン (著), 徳高 平蔵, 岸田 悟, 藤村 喜久郎 (訳), 自己組織化マップ,Springer, (1996).

[Ohkita 08] 大北 正昭, 徳高 平蔵ほか: 自己組織化マップとそのツール, Springer Japan, (2008).