

# プログラム作成時の行き詰まり検出のための 主観的作業負荷推定の試み

松田 力<sup>†1</sup> 角田 雅照<sup>†1</sup>

**概要**: プログラミングの実習において、学生の行き詰まりを発見することは重要な課題である。これまで実習における学生の行き詰まりを検出するための方法がいくつか提案されているが、本研究では学生の感情（表情）から主観的な作業負荷を推定し、それに基づきプログラミング作成の行き詰まりを検出することを試みる。

## 1. はじめに

プログラミングの実習において、学生の行き詰まりを発見することは重要な課題である。プログラミングの実習では、学生にプログラムを作成したり修正したりする課題を課すが、全ての学生がスムーズに解にたどり着くということはない。何人かは作成、修正の途中で行き詰まるということが起こりうる。もちろん、これらの行き詰まりにおいて、学生自身が自力で解決することは重要である。一方で、長時間行き詰ったままの学生も存在し、そのような場合については、教員やティーチングアシスタントが手助けするほうがよく、それにより学生は新たな課題に取り掛かることができる。

そのため、これまで実習における学生の行き詰まりを検出するための方法がいくつか提案されている[1]。ただし、それらは主に学生のコンピュータの操作や、学生の作成したプログラム（成果物）の変化に着目していた。本研究では学生の感情（表情）から主観的な作業負荷を推定し、それに基づきプログラミング作成の行き詰まりを検出することを試みる。プログラマーの脳などの生理活動からプログラマーの作業状態を推定する試みは存在するが[2]、表情に基づいて推定する試みは、我々の知る限り存在しなかった。

## 2. 学習者の感情計測

本研究では、学習者の感情計測のためにシステムを作成した。作成したシステムは、パソコンに内蔵のカメラや Web カメラを用いて、人間の顔画像を撮影し、その画像の表情から感情を読み取り、数値として感情を表すものである。パソコンの正面に座っているユーザーの写真を秒毎に撮影し、その時間毎の表情を読み取り、感情の数値と表情を読み取った時間の記録をテキストファイルにログとして保存する。システムを使用した際に保存するテキストのログのイメージを図 1 に示す。

本システムでは、Microsoft が提供する Web API である、Cognitive Services の Emotion API を用いた。この API により、撮影した画像から人間の顔画像を特定し、かつそれらの顔画像から表情を読み取り数値として感情を表すことが

```
2017/10/10 15:14:41neutral 0.997
sadness 0.003
surprise 0.0
contempt 0.0
disgust 0.0
2017/10/10 15:15:00happiness 0.424
neutral 0.359
sadness 0.172
contempt 0.026
disgust 0.015
```

図 1 出力するログのイメージ

可能となっている。Emotion API は anger, contempt, disgust, fear, happiness, neutral, sadness, surprise の 8 種類の感情を表すことが可能であり、感情が強いものほど高い数値で表される。

システムでは、この API を利用しつつ、下記の手順を一定時間ごと（実験では 5 秒ごと）に繰り返してログを記録する。

1. パソコンの内蔵カメラで撮影して画像を保存する。
2. 保存した画像を API に渡す。
3. API から得られた結果をテキストファイルにログとして記録する。

システムは Mac OS 上で実装し、ノートパソコンの内蔵カメラを用いることとした。システム実装のためのプログラミング言語として Python を用いた。システムを実装するにあたり、表情を読み取るための Emotion API を使用する必要があったため、Microsoft Azure のアカウント作成を行った。撮影した画像に顔が含まれているかの判定を行う際に Open CV を使用した。そのため、プログラムを動作させるためにはライブラリのインストールが必要となる。

なお、感情の認識方法については、上記 API や Intel 社が提供する RealSense Camera (<https://www.intel.com/content/www/us/en/support/articles/000006099/emerging-technologies.html>)などの既存の手法を用いる他に、独自に画像の感情認識をするモデルを作ることも可能である。顔の画像と、それに対応付けられた前述の 8 種類の感情の数値データが、Kaggle において fer2013 として公開されており、これを用いることにより比較的容易にモデルを作成することができる。

<sup>†1</sup> 近畿大学  
Kindai University

表 2 質問票の一部[3]

Q1	やさしかった	3	2	1	0	-1	-2	-3	難しかった
Q2	単純だった	3	2	1	0	-1	-2	-3	複雑だった
Q3	頭を使う必要はなかった	3	2	1	0	-1	-2	-3	頭を使う必要があった
Q4	大ざっぱでよかった	3	2	1	0	-1	-2	-3	正確さが要求された
Q5	体を使う必要はなかった	3	2	1	0	-1	-2	-3	体を使う必要があった
Q6	楽な作業だった	3	2	1	0	-1	-2	-3	きつい作業だった

	0.005	0.002124
	anger	contempt
14:24:57	0	0.005
14:25:03	0	0.001
14:25:10	0	0.007
14:25:17	0	0.024

図 2 作業中の感情値の平均値の計算例

表 1 採用された変数と絶対誤差

目的変数	採用された説明変数	絶対誤差
精神負担	happiness	1.06
身体負担	anger, neutral	2.09
時間圧力	happiness, fear	3.48
作業成績	contempt, disgust	1.23
努力	fear, anger	0.47
フラストレーション	disgust, surprise	1.47

### 3. 実験

実験では、情報科学を専攻する学部 4 年生 5 人の被験者に、プログラムをフルスクラッチで作成する課題を与え、課題に取り組んでいるときの感情を計測した。課題は会津大学のプログラミング問題のオンライン採点システムから 1 問を選択した。回答時間は 30 分とし、用いる言語は被験者が慣れている Java とした。インターネットで Java の文法などを調べることは許可した。被験者全員が制限時間内にプログラムの作成を完了させることができなかった。

主観的な作業負担を計測するための非常に有名なものとして、NASA-TLX があるが、本研究では、これをより使いやすくした質問票[3]を用いた。これは精神負担、身体負担、時間圧力、作業成績、努力、フラストレーションを計測するための質問票であり、表 1 のような質問票の回答に基づきそれぞれの値が 10 から -10 と算出される。数値が大きいほど負担が大きいことを示す。

次に、収集した表情のデータから、被験者の作業負担を重回帰分析により推定することを試みた。説明変数は 2 章で説明した 8 種類の感情の作業中の平均値（図 2 の赤枠部分）とし、変数選択により変数を絞り込んだ。用いるデータ数が少ないため、採用する説明変数は最大で 2 個とした。

目的変数は精神負担、身体負担、時間圧力、作業成績、努力、フラストレーションの 6 種類とした。目的変数の値の予測時にはリーブワンアウト法を用い、予測精度の評価

には予測値と実測値との絶対誤差を用いた。

予測結果を表 2 に示す。時間圧力が最も予測誤差が大きく、身体負担も誤差が 2 を超えていたが、その他の目的変数では予測誤差が比較的小さかった。このことから、本システムに基づき主観的な作業負担をある程度推定可能であるといえる。

実験ではどの被験者も完成していなかったが、「順調に進んでいた」と答えた学生のフラストレーションや時間圧力は小さかった。そのため、成果物のみで判断するよりも、学生の状況をより判断しやすい可能性がある。また、時間圧力は誤差が大きかったため、フラストレーションの予測値に基づき、行き詰まりを判断すると良い可能性がある。

### 4. おわりに

本研究では、作業負担の高い学生は学習に行き詰まっていると想定し、プログラミングの学習時に、学生の表情に基づいて、主観的作業負担を推定する方法について検討した。実験では、主観的作業負担をある程度高い精度で予測することができた。成果物だけでは行き詰まりの判定が容易でなかった可能性があり、フラストレーションに着目したほうが、行き詰まりを特定しやすい可能性がある。

**謝辞** 本研究の一部は、日本学術振興会科学研究費補助金（基盤 A：課題番号 17H00731）による助成を受けた。

### 参考文献

- [1] 杉山匠, 小暮悟, 野口靖浩, 山下浩一, 小西達裕, 伊東幸宏: プログラミング演習における迷い検出に基づく学習教育支援システムの構築, 電子情報通信学会技術研究報告, vol.116, no.438, pp.81-86 (2017).
- [2] B. Floyd, T. Santander, and W. Weimer: Decoding the representation of code in the brain: an fMRI study of code review and expertise, In Proc. of International Conference on Software Engineering (ICSE), pp.175-186 (2017).
- [3] K. Shinohara, H. Naito, Y. Matsui, and M. Hikono: The effects of “finger pointing and calling” on cognitive control processes in the task-switching paradigm, International journal of industrial ergonomics, vol.43, no.2, pp.129-136 (2013).