プログラミングコンテストシステムへの提出履歴データとその分析

堤 祥吾[1]　楊 欣[1]　崔 恩瀞[2]　井上 克郎[1]

概要：本論文では，プログラミングコンテストシステムから参加者の提出履歴を取得して構築できるデータセットについて紹介する．また，本データセットを利用してプログラム特徴分析を行う方針を述べる．

キーワード：プログラミングコンテスト, Codeforces, データセット

1. プログラミングコンテスト

この章では，本データセットの作成のために参照するプログラミングコンテスト [4] について説明する．プログラミングコンテストには複数の種類があるが，本論文ではアルゴリズムに関する問題を解く種類のコンテストについて述べる．

以下では，プログラミングコンテストに用いられるオンラインジャッジシステム (OJS) とそれを用いたプログラミングコンテストについて説明する．

1.1 オンラインジャッジシステム (OJS)

プログラミングコンテストにおいて，その採点に利用されているオンラインジャッジシステム (以下 OJS) について説明する．OJS は，利用者に問題を提示し，利用者から受け取った回答の正誤を通知するシステムである．一例として，国内で代表される AIZU ONLINE JUDGE[*1]，アメリカの Topcoder[*2]が存在する．

OJS は問題として，問題文，サンプルテストケース，実行時間やメモリ制限等の実行上の制約を与える．利用者は制約を満たすプログラムを作成し，そのソースコードをシステムに提出する．システムは提出ソースコードをコンパイルし，予め用意されている複数のテストケースを実行する．実行後，テストケースを通過すれば正解を，間違った出力や実行制限を違反した場合はその旨を利用者に通知する．

[1] 大阪大学
Osaka Uniersity
[2] 奈良先端科学技術大学院大学
Nara Institute of Science and Technology

[*1] http://judge.u-aizu.ac.jp/onlinejudge/
[*2] https://www.topcoder.com/

名前	AU	提出履歴	API
Codeforces	29927	○	○
Topcoder	3675	△[*5]	○
CodeChef[*6]	12129[*7]	○	×

表 1　プログラミングコンテストサービスの特徴[*4]

1.2 プログラミングコンテスト

プログラミングコンテストは，OJS 上で複数の参加者が同じ問題セットを解く方式で行われる．正解問題数や回答時間に応じて参加者の順位が決定する．順位に応じて参加者の rating[2] が変動し，参加者の熟練度がわかるようになっている．

2. 作成するデータセットの概要

2.1 データセットを構築するために利用するサービス

プログラミングコンテストサービスは複数あり，国内外でよく利用されているものの一部を表 1 にまとめた．表の各項目について説明する．「AU」は過去 6 ヶ月以内に 1 度でもコンテストに参加したことがあるユーザー (アクティブユーザー) の総数を表す．「提出履歴」は，提出されたソースコードにアクセス可能かを表す．「API」は，主にユーザーや提出履歴の取得に用いる API が存在するかどうかを表す．

本データセットを構築するにあたっては，大手プログラミングコンテストサービスである Codeforces を利用する．選定理由としては，他のサービスと比較して，アクティブユーザー (以下 AU) が多いこと，提出履歴が公開され，アクセスしやすいこと，API の有無が挙げられる．

[*5] 各問題に対する最終提出のみ閲覧可能
[*6] https://www.codechef.com/
[*7] 過去 6 ヶ月以内に 1 度でも提出したことがあるユーザー数

2.2 データセットに含まれるデータ

データベース上に構築した本データセットの内容を表 2 に示す．Codeforces の AU のうち，3000 人をサンプリングし「ユーザー情報」テーブルに格納した．また，サンプルユーザーが過去 6 ヶ月以内に提出したソースコードを「ソースコード情報」のテーブルに格納した．

「ユーザー情報」に含まれる情報のうち，「最大 rating」は過去に到達したことがある rating の最大値を，「ソースコード数」は過去 6 ヶ月の間に提出したソースコード数を表している．「ソースコード情報」に含まれる情報のうち，「問題 ID」は提出ソースコードに対応する問題番号を，「提出結果」は問題に正答したソースコードかどうかを表している．

2.3 統計情報

図 1 は，データセット中のソースコード長の度数分布を表す．縦軸はソースコード数，横軸は長さ (単位は byte) である．提出ソースコード長の平均は 1256byte であった．本データセットに含まれるソースコードは，開発で作成されるソースコードより短いことがわかる．

図 2 は，データセット中の修正提出数の度数分布を表している．ここで修正提出数とは，ある同一の問題に対して，最初の正解までに提出した不正解提出の数と定義する．このグラフでは，縦軸を提出数，横軸を修正提出数とする．提出の多くは 1 度目で正解しており，修正の増加とともに分布が減少している．

テーブル種別	データ数	内容
ユーザー情報	3000	ユーザー名，rating， 最大 rating，ソースコード数
ソースコード情報	394521	問題 ID，プログラミング言語， 提出者，提出結果，ファイルパス名

表 2 データセットの内容

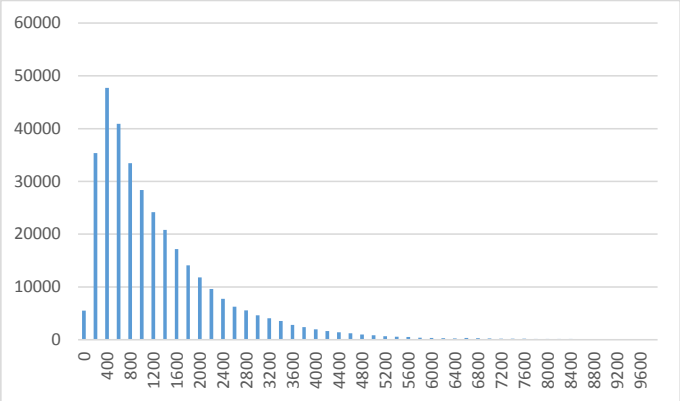

図 1 提出ソースコード長の分布

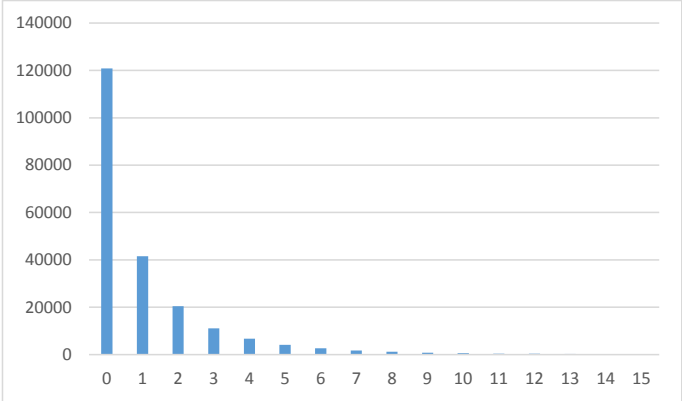

図 2 修正提出数の分布

3. 分析計画

3.1 分析の目的

プログラミング上級者の特徴を発見することは，学習者が上級者の持つプログラミング技術を理解し，自身の能力向上に役立てることができるという観点から教育上有用である．また，プログラミングコンテストにおいては，能力を計測する定量的尺度の 1 つとして rating が用いられており，参加者の能力を比較することが容易である．そこで，rating が高い参加者共通の特徴を抽出することにより，学習者や教育者に対して有用な結果を得ることを分析の目的とする．

3.2 分析の方針

分析の方針として，提出時間や提出回数，rating が，ソースコードの特徴とどのような関連があるかを調べることとする．ソースコードの特徴量としては，ソースコードの長さやコードクローン [1]，コードスメル [3] などが考えられる．またプログラミングコンテストにおいては自作データ構造ライブラリやマクロの多用等の特有の特徴も存在するため，それらを考慮した特徴量を扱う必要がある．

参考文献

[1] Ira D Baxter, Andrew Yahin, Leonardo Moura, Marcelo Sant'Anna, and Lorraine Bier. Clone detection using abstract syntax trees. *ICSM '98 Proceedings of the International Conference on Software Maintenance*, p. 368, 1998.

[2] Open codeforces rating system, 2015. http://codeforces.com/blog/entry/20762.

[3] Martin Fowler, Kent Beck, John Brant, William Opdyke, and Don Roberts. *Refactoring:Improving the Design of Existing Code*. Addison Wesley, 1999.

[4] 秋葉拓哉, 岩田陽一, 北川宜稔. プログラミングコンテストチャレンジブック. マイナビ出版, 2010.