深層学習を用いたソースコード分類のための学習用データセット改善手法の提案

藤原裕士[1] 崔 恩瀞[2] 吉田則裕[3] 井上克郎[1]
[1]大阪大学 [2]京都工芸繊維大学 [3]名古屋大学

研究背景：ソースコード分類

- 例：ソースコードの機能別分類
  - ➢ 大規模ソフトウェアリポジトリに新規登録されたソースコードに対して機能タグを自動付与
- 既存ソースコードの検索や再利用の効率化に貢献

研究背景：深層学習における不均衡データ問題

- 分類クラス間におけるデータ数の不均衡は訓練に悪影響
  - ➢ データ数が少ないクラスを無視する傾向

データ数の比9:1のデータセット

クラス1	クラス2

- ➢ 各クラスを正答率0.8で分類→分類精度0.8
- ➢ 全てのデータをクラス1に分類→分類精度0.9

- 不均衡データの対処法は様々
  - ➢ 例：ランダムサンプリング，重みのクラス別設定

<問題点>

深層学習を用いたソースコード分類の既存研究で，データセット構築に関する工夫があまりなされていない

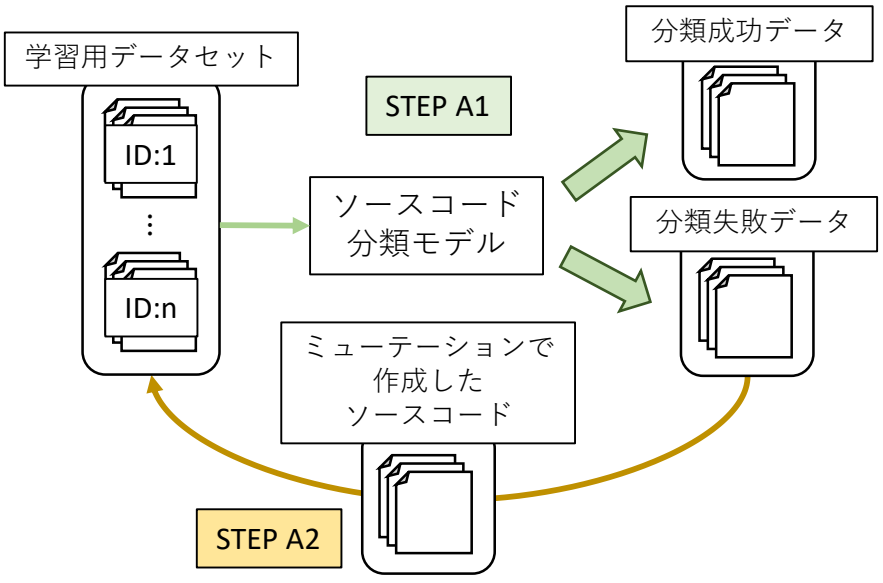
- ➢ 多くの場合，ランダムサンプリングのみ
- ➢ 学習用データセットの構成次第でさらに良い分類精度を実現できる可能性がある

提案手法：動的な学習用データセット改善手法

本研究ではソースコード分類のための手法を提案

STEP A1: 深層学習モデルの訓練を行い，分類精度を評価
STEP A2: 深層学習モデルの訓練結果に応じて，新たな学習データ(ソースコード)を作成し，学習用データセットに追加
- ソースコードの作成にはミューテーションを利用

STEP A3: STEP A1・A2の繰り返し



分類に失敗したクラスのデータを増やすことで，モデルの訓練の際にそのクラスが無視されにくくなる

評価実験

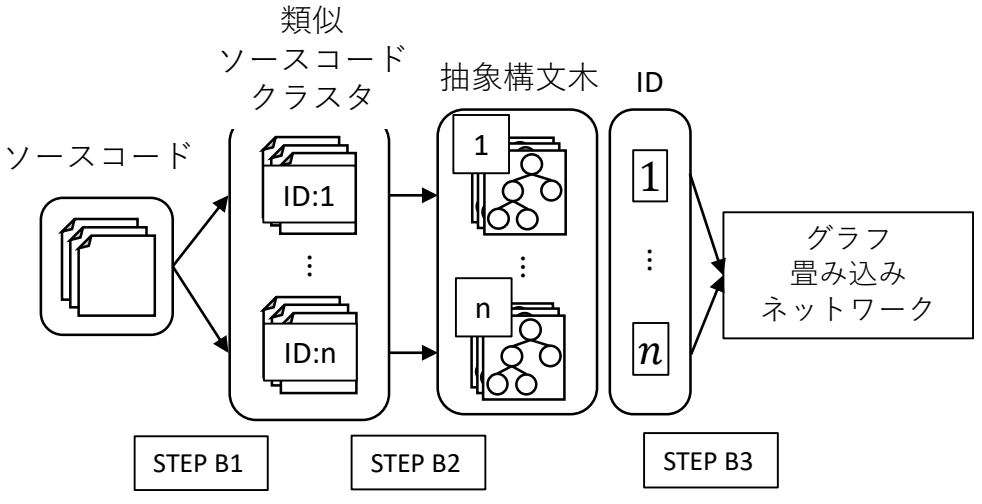
- ベースライン手法とする2種類の不均衡データ対処法と提案手法の計3種類の手法からデータセットを構築
- 構築したデータセットを用いて訓練された分類モデルの分類精度を比較

評価実験で用いるソースコード分類手法

グラフ畳み込みネットワーク[1]を利用

モデル訓練手順（STEP B）
STEP B1: 類似ソースコードをクラスタリングし，クラスタ毎に固有のIDを割当
STEP B2: 各ソースコードを抽象構文木に変換
STEP B3: 抽象構文木を説明変数，IDを目的変数とした教師あり学習を実行

➢ ソースコードの抽象構文木を入力するとそのソースコードが属する可能性の高いクラスタのIDを推測するモデルを作成



評価実験に使用するデータセット

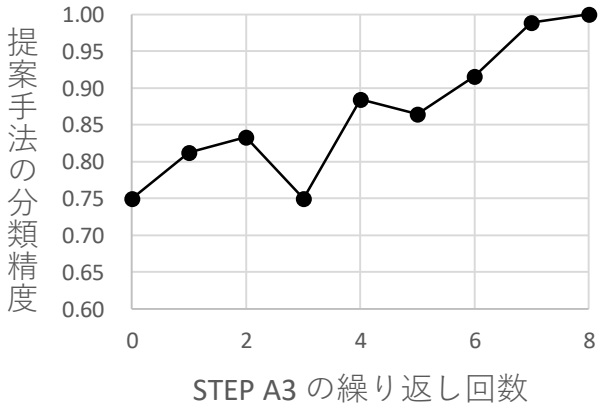
OpenSSL 0.9.1~1.1.0*1を利用したデータセット
- ➢ 構文的に類似したソースコードのクラスタ20個

ベースライン手法

- Method-oriented
  - ➢ 各クラスタに含まれるメソッド数が均等
- Node-oriented
  - ➢ 各クラスタに含まれるノード数が均等

実験結果

手法	分類精度
Method-oriented	0.92
Node-oriented	0.94
提案手法	1.00



提案手法を用いて構築したデータセットで訓練したモデルはベースライン手法よりも高い分類精度

今後は，別のデータセットや別のソースコード分類手法に本手法を適用し，有効性を評価する予定

[1] Kipf, T. N. and Welling, M.: Semi-Supervised Classification with Graph Convolutional Networks, Proc. of ICLR 2017 (2017)
*1 https://www.openssl.org/

本研究は JSPS 科研費 18H04094 の助成を受けたものです。