

機械学習を用いた コードクローンの危険予測手法

今里 文香 堀田 圭佑 肥後 芳樹 楠本 真二

コードクローンの有害、無害

既存研究によると…

- コードクローンの36~38%は、一貫した修正を要する(危険)
- コードクローンの48%は、生成後1度も変更されない(無害)



本研究の目的

危険な(有害になり得る)コードクローンを機械学習で予測する

機械学習

- 危険なコードクローンの定義は？
- 何を特徴量とする？
- どのように学習、予測する？

先行研究(1)

有用なコードクローンを予測する手法

有用なコードクローンは開発者ごとに異なるとして、
あらかじめ開発者自身が有用かどうかを分類しておく



8人の被験者実験の結果、70%以上の精度で予測できた

先行研究(2)

コードクローン生成直後にそのコードが危険か予測後に1度でも一貫した変更が加えられるものを危険と定義



危険でないと判定したもののうち、95%以上が真であった
危険なコードクローンのうち、34%を検出できた

提案手法の特徴

- バグを起こしたことのあるクローンセットの系譜を危険と定義
- 危険な系譜とそうでない系譜の特徴(傾向)を学習し、**現在のコード**に存在するクローンセットの危険を予測する
- 多くの既存手法が学習と予測のどちらにもソフトウェア開発履歴(系譜)を使うのに対して、提案手法は**予測時にはソフトウェア開発履歴が不要**なので、より高速に予測できる

提案手法の手順

1. CRDの類似度に基づいてクローンセットの系譜を取得
2. 危険な系譜かどうかを判定
3. 訓練データの特徴量から学習モデルを構築

1. クローンセットの系譜を取得

CRD

コードクローンを構成するコード片のおおよその位置情報を表す
表現形式

- CRDの類似度を考慮すると、コード片の位置が変わるような修正が加えられた場合でも、コードクローンを追跡できる

2. 系譜の危険判定

- 開発過程においてバグ修正が行われた系譜を危険と定義
- 変更されたことのあるクローンセットを含むリビジョンのコミットメッセージを参照して、バグ修正が行われたとわかったら、その系譜を危険と判定する
- バグ修正の判定は、論文「When do changes induce fixes?」の判定パターンを利用する

3. 訓練データの特徴量

- その時点のクローンセットから得られる特徴量だけを使った
(系譜は不要)
- クローンセットの要素数、トークン数の平均値、ファイル数など
- 全30種類
- 詳細は不明

評価実験

表 2 実験データの内訳

プロジェクト名	危険	危険でない
jEdit	1,695	2,563
Ant	2,835	22,818

評価結果

表 5 評価結果

アルゴリズム	jEdit		Ant	
	Precision	Recall	Precision	Recall
J48	0.93	0.89	0.67	0.45
RandomForest	0.92	0.89	0.60	0.41
ベイジアンネットワーク	0.83	0.90	0.45	0.54
SVM	0.95	0.83	0	0
ロジスティック回帰	0.92	0.86	0.28	0.03

考察・まとめ

- 危険なクローンセットの傾向は、プロジェクトによって異なる部分がある
- 特徴量に「要素を含むファイルのディレクトリ上における広がり」が有効な可能性があることがわかった
- その他、ノイズとなる特徴量を選別する必要がある