

Measuring Software Redundancy

ソフトウェアの冗長性を客観的に測定するために、改めて定式化を行い、測定手法を提案した。さらに実験を行い、手法の有用性を示した。

ICSE' 2015

Antonio Carzaniga | Andrea Mattavelli
Mauro Pezze

背景

- ❖ ソフトウェアの冗長性の重要性
 - フォールトトレラント
故障が起きてもバックアップシステムを使うなどして動作に支障がない
 - 信頼性向上
- ❖ システムの冗長性を明確に測る方法がない
 - いくつの冗長性がある？
 - その冗長性は同じアルゴリズムを使っている？
 - Semantic Clone

本論の目的

- ❖ ソフトウェアの冗長性の意味を定義
- ❖ 冗長性測定の手法を確立したい
 - Automatic mining of functionally equivalent code fragments via random testing(ISSTA' 09)
- ❖ コードの冗長性とアルゴリズムの冗長性を区別
 - **high-level redundancy**
違うアルゴリズムを実装してある冗長性
 - **low-level redundancy**
同じアルゴリズムだが、違うコードで実装してある冗長性

コードフラグメント

<i>linkage:</i> int x; int y;	
int tmp = x; x = y; y = tmp;	x ^= y; y ^= x; x ^= y;
<i>linkage:</i> AbstractMultimap map; String key, Object value;	
map.put(key, value);	List list = new ArrayList(); list.add(value); map.putAll(key, list);

Fig. 1. Examples of redundant code fragments.

- ❖ コードフラグメント
コードの断片. これ比較して, 値や変数が違うコードだと冗長性があると判定.
- ❖ 測定方法
テストを実行してあらゆる変数などを出力し, 比較する.

実験

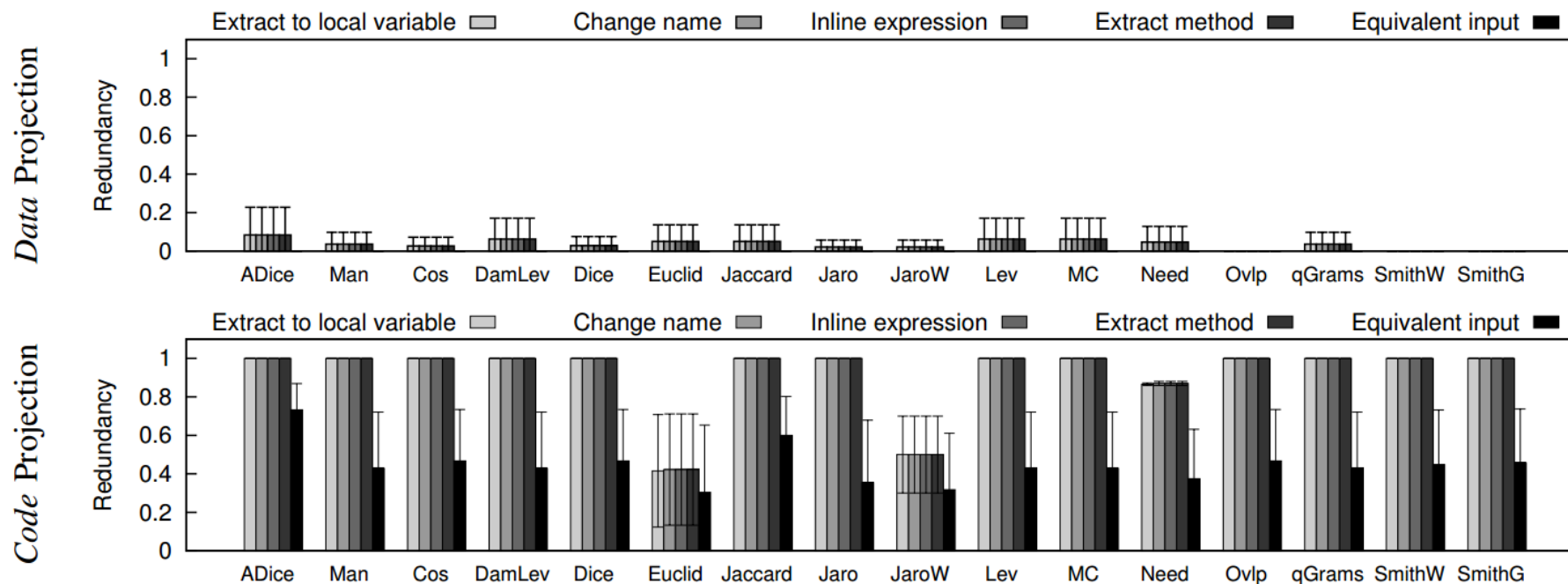
❖ 妥当性を評価

- 自身を冗長性と判断してしまわないか.
- リファクタリングを行って, 冗長性と判断されるか.
- わざと違う結果を返すよう改変して, 冗長性と判断しないか.

❖ 冗長性のレベルの評価

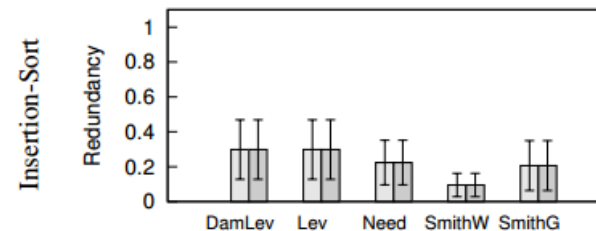
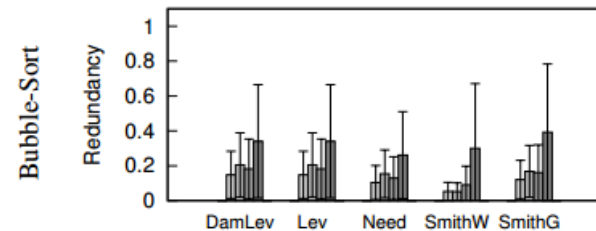
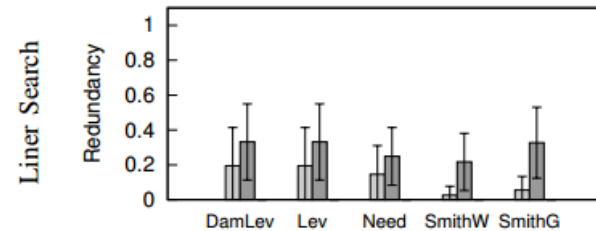
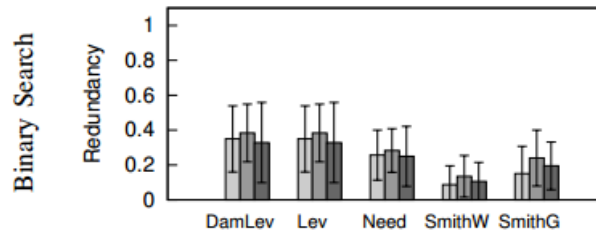
- High-level redundancyとLow-level redundancyの違いを検出できるか.

妥当性を評価

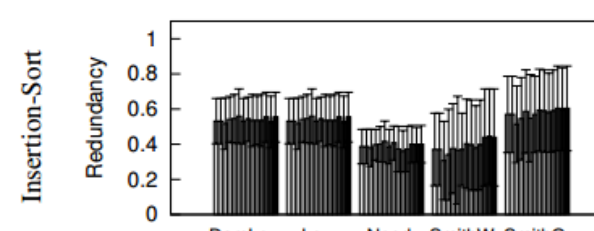
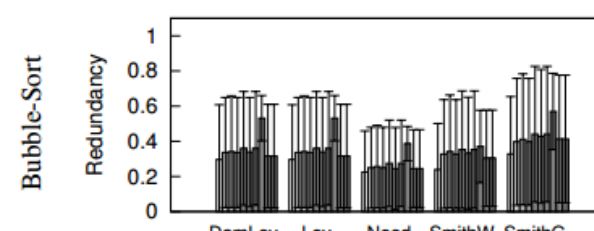
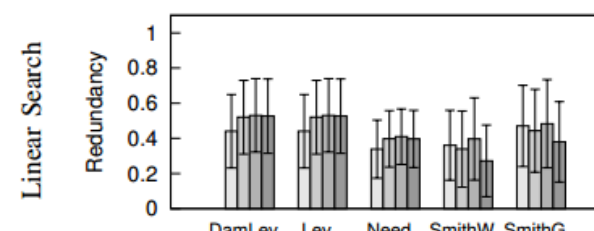
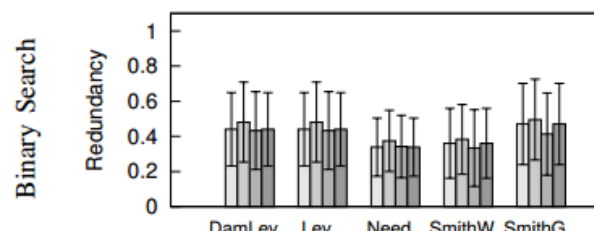


- ❖ Data Projection (自己検出率) は低く, Code Projection (リファクタリング検出率) は高い結果になった.
- ❖ 正しく判別できていると言えそう

冗長性レベルの評価



同じアルゴリズムでの冗長性評価



違うアルゴリズムでの冗長性評価

❖ 精度は高くないが、ある程度判別できた。

❖ 判別の精度よりも、冗長性の定義づけを重視

まとめと所感

- ❖ ソフトウェアの冗長性を定式化
 - 自己検出
 - リファクタリング
- ❖ 冗長性についての実験を行った
 - コードフラグメント
 - テキスト解析など使っていない
- ❖ 所感
 - 冗長性を定式化したのは素晴らしい.
 - 決まったパターンでしか実験を行っていないので, もう少し追加検証を見てみたい.
 - さらに検出精度を上げられるとおもしろいかも.