

Use of Relative Code Churn Measures to Predict System Defect Density

Nachiappan Nagappan, Thomas Belle

(コードチェーンの相対測量を使用して欠陥密度
予測モデルの構築)

コードチャーン (Code Churn) とは?

- ソフトウェアユニット内で追加、削除または変更されたコードの量

背景

- ソフトウェア組織には,早期に潜在的な欠陥を警告するシステムがあったら,便利である
- 前の研究によると,コードの行数 (Lines of Code, LOC) のような絶対測量は,商用ソフトウェアシステムの予測変数としては不十分
- 一般に,変更履歴に基づくプロセス測量は,より良い指標
- システムソフトウェアの欠陥密度を予測するために、コードチェーンを使用

目的

- コードチェーンの相対メトリクスを用いて
欠陥密度の予測モデルの紹介

研究メソッド

- コードチェーンは,バージョン管理システムによって自動的に記録されるため,システムの変更履歴から簡単に抽出できる
- 対象: Windows Server 2003 (W2k3)
 - W2k3のリリースとW2k3 Service Pack 1 (W2k3-SP1)のリリースの間のコードチェーンを分析し, W2k3-SP1の欠陥密度を予測
 - 2465 バイナリが使用された

コードチェーンの絶対メトリクス

- 合計LOC
- 変更されたLOC
- 削除されたLOC
- ファイル数
- 変更する時の時間
- 変更の数
- 変更されたファイル

コードチェーンの相対メトリクス

- M1～M8のメトリクスを使用
 - M1: チェーンされたLOC /合計LOC
 - チェーンされたLOC → 追加LOC + 変更LOC
 - M2: 削除されたLOC /合計LOC
 - M3: チェーンされたファイル/ファイル数
 - M4: チェーンの数/チェーンされたファイル
 - M5: チェーンするためかかる時間/ファイル数
 - M6: 編集された行数/チェーンするためかかる時間
 - M7: チェーンされたLOC /削除されたLOC
 - M8: 編集された行数/チェーン数

研究の仮説

1. コードチェーンの相対測量(コードの相対変化量)の増加に伴い, システムの欠陥密度も増加
2. システムの欠陥密度を説明するために, コードチェーンの絶対測量より相対測量を使用する方が良い
3. コードチェーンの相対測量は, システム欠陥密度の効率的な予測変数として使用できる
4. 相対コードチェーン測量を使用して, バグのあるバイナリとバグのないバイナリが区別できる

相関分析

Table 2. Cross Correlations. All correlations are significant at the 0.01 (99%) level (2-tailed).

		M1	M2	M3	M4	M5	M6	M7	M8	Defects /KLOC
M1	ρ	1.000	.834	.795	.413	.707	.651	.466	.588	.883
M2	ρ		1.000	.645	.553	.747	.446	.219	.492	.798
M3	ρ			1.000	.186	.749	.434	.445	.269	.868
M4	ρ				1.000	.531	.429	.210	.631	.288
M5	ρ					1.000	.263	.201	.390	.729
M6	ρ						1.000	.701	.843	.374
M7	ρ							1.000	.507	.288
M8	ρ								1.000	.262
Defects/ KLOC	ρ									1.000

- コードチェーンの相対測量と欠陥密度の相関関係进行分析
- 表2は, 相対測量と欠陥/ KLOCの間に正の関係が存在することを示す
- コードチェーンの相対測量が増加すると, システムの欠陥密度も増加

モデル構築

- 相対メトリクスを使用して作成された予測モデル（相対モデル）と絶対メトリクスを使用して作成された予測モデル（絶対モデル）を比較
- 絶対モデル：
 - 欠陥/KLOCは従属変数であり, 独立変数は絶対メトリクス
- 相対モデル：
 - 欠陥/KLOCは従属変数であり, 独立変数は相対メトリクス

モデル構築

Table 5. Relative Measures Model Fits

Model	R^2	Adjusted R^2	F-Test sig.
All measures	0.811	0.811	1318.44, ($p < 0.0005$)
Step-wise regression	0.811	0.811	1507.31, ($p < 0.0005$)
PCA	0.749	0.748	2450.89, ($p < 0.0005$)

- 絶対測量の R^2 の値が低い
 - すべての独立変数を使用した多重回帰 : 0.052, 段階的回帰 : 0.051 ,
PCA : 0.026
- 3つの結果（すべての独立変数を使用した多重回帰, 段階的回帰とPCA）に基づいて, 絶対測量は良い予測変数ではない
- 一方, 相対測量の方が良い予測変数

欠陥密度の予測モデル

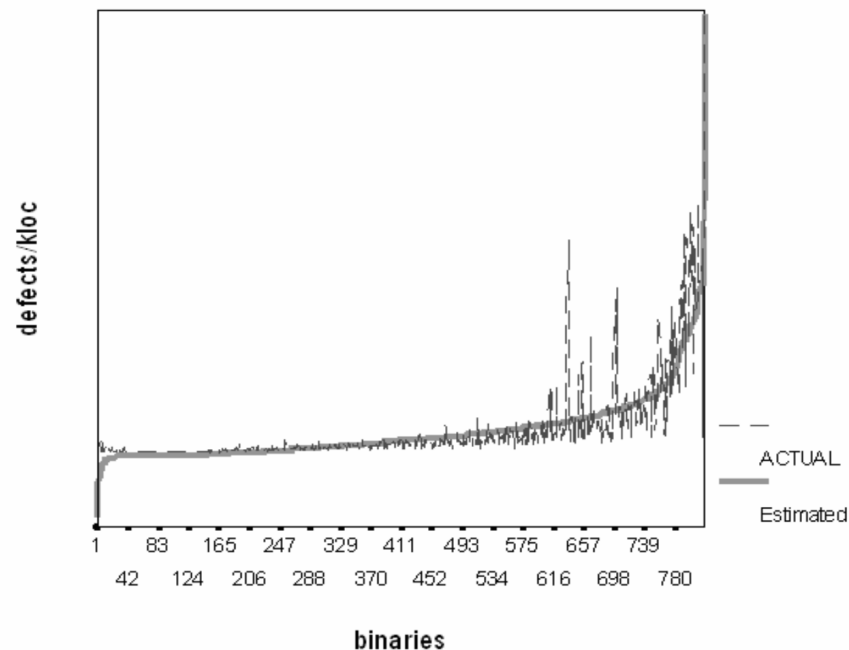


Figure 3: Actual vs. Estimated System Defect Density

- モデルの予測能力を確認する
- バイナリの2/3 (1645) をランダムに選択し, 予測モデルを作成
- 残りの1/3 (820) を使用し, 予測の正確度を検証
- 図3は, 推定および実際の欠陥密度を示している

欠陥密度の予測モデル

Table 7. Correlation Results

Model	Pearson (sig.)	Spearman (sig.)
All measures	0.889 (p<0.0005)	0.929 (p<0.0005)
Step-wise regression	0.889 (p<0.0005)	0.929 (p<0.0005)
PCA	0.849 (p<0.0005)	0.826 (p<0.0005)

- 推定値と実際の値の相関を分析
- **正の相関係数**は, 実際の欠陥密度の増加に伴い, 推定欠陥密度も増加することを示す
- システムの欠陥密度を予測するために, コードチェーンの相対測量が使用できると言える

判別分析

Table 11. Discriminant Analysis

Model	For Model Fit (for 1645 binaries to build the model)		For Test Data (820 binaries)
	Eigen value	Classification ability	Classification ability
All Measures	1.063	1464/1645 (90.0%)	735/820 (89.6%)
PCA	0.601	1461/1645 (88.8%)	739/820 (90.1%)

- 判別分析を使用し, バイナリーにバグが発生しやすいものまたは発生しにくいものと識別
- 相対メトリクスを用いる予測モデルは、正確度が89%
- コードチェーンの相対測量を使用して, バグが発生しやすいバイナリとバグが発生しにくいバイナリが区別できる

結論

- 欠陥密度を予測するために, コードチェーンの相対メトリクスが良い予測変数であること
- ケーススタディから,
 - コードチェーンの相対測量 (変化量) を増加すると, システムの欠陥密度も増加
 - 欠陥密度を説明するために, コードチェーンの絶対測量より相対測量を使用する方が良い
 - コードチェーンの相対測量は, システム欠陥密度の効率的な予測変数である
 - コードチェーンの相対測量を使用して, バグが発生しやすいバイナリとバグが発生しにくいバイナリが区別できる

所感

- コードチェーンの説明が不足
- Code Churn(コードチェーン)
 - ソフトウェアの変更の度合いを表すメトリクスである
 - 2つのリビジョン間で単純にコード量の差を測るのではなく、実際に追加, 変更, 削除された量を計測することでソースコードの変化量と不具合との関連を調べたものである
 - それをもとに Nagappan が RelativeCodeChurnを提案し, コード行数以外にもいくらかのメトリクスをあげ, それらの絶対測量を相対測量にすることで, 欠陥密度と深く関連していると発表している

—出原 真人, “オープンソースソフトウェア開発における 版更新と不具合出現の
関連分析”, 卒業研究報告, 京都工芸繊維大学, 2011