
Who Should Fix This Bug?

ICSE '06

John Anvik | Lyndon Hiew | Gail C. Murphy

発表: 松尾 優太

BackGround

- ❖ OpenBugRepositories
開発者とユーザーがバグや機能拡張を投稿.
Bugzilla, GNATS, JIRAなど.
- ❖ The Life-cycle of a Bug Report

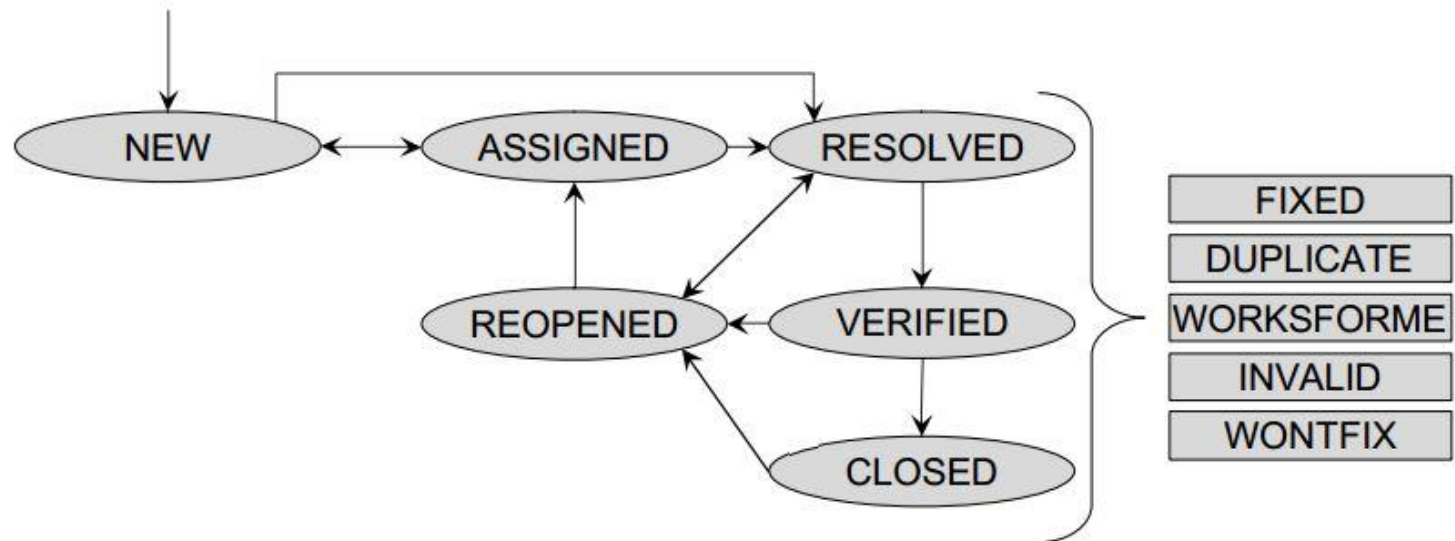


Figure 1: The life-cycle of an Eclipse bug report.

BackGround - Bug Triage

❖ Triageとは

“対応人員や物資などの資源が通常時の規模では対応しきれないような非常事態に陥った状況で、最善の結果を得るために、対象者の優先度を決定して選別を行うこと。”

Wikipedia

❖ OSS開発の現状

- 毎日、数百件～数千件のバグ報告.
- 不足しているコミッター
- バグの選別が必要

機械学習アルゴリズムを用いて、
バグをトリアージしよう！

BackGround - Macine Learning

❖ Text Categorization

テキストをカテゴリ分けするための機械学習.

BugReport→Text(Instances) Developer→Category

Instancesはカテゴリを表すLabelを持っているのでは！？

➤ Training Classifier

Instancesと既知のLabelを入力にして,

Classifier(分類子)を作る.

A SEMI-AUTOMATED APPROACH TO BUG ASSIGNMENT

❖ バグ割り当ての4ステップ

- Characterizing Bug Report.
- Labeling Bug Report.
- Selecting Training Reports.
- Applying a Machine Learning Algorithm.

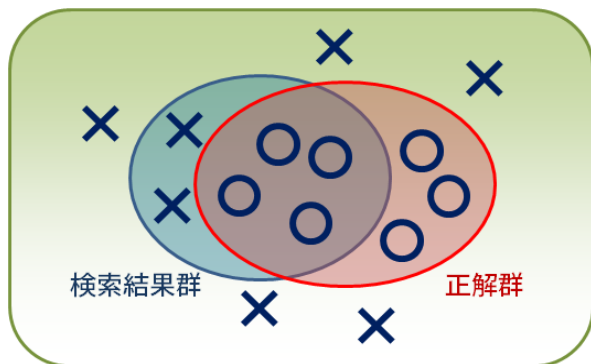
❖ PrecisionとRecall

- Precision(適合率)
 - Recall(再現性)
-

Precision (適合率) と Recall (再現率)

$$\text{Precision} = \frac{\text{\# of appropriate recommendations}}{\text{\# of recommendations made}}$$

$$\text{Recall} = \frac{\text{\# of appropriate recommendations}}{\text{\# of possibly relevant developers}}$$



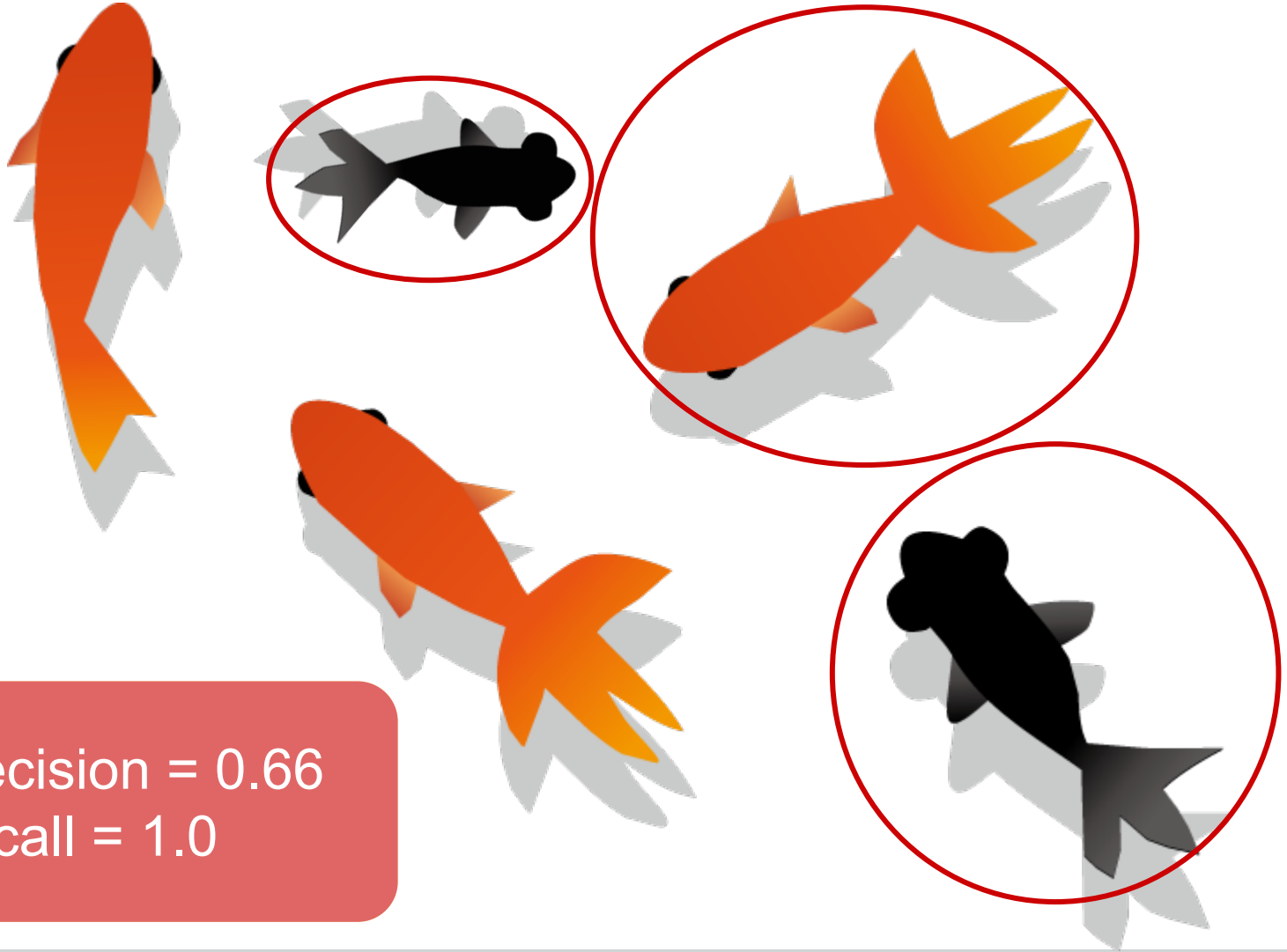
❖ **Precision**
予測したうち, 正解の割合

適合率 Precision = $\frac{4}{6} = 0.67$

❖ **Recall**
正解集合のうち, 正解の割合

再現率 Recall = $\frac{4}{7} = 0.57$

Recallのこと—金魚の例



Precision = 0.66
Recall = 1.0

A SEMI-AUTOMATED APPROACH TO BUG ASSIGNMENT

❖ SEMI-AUTOMATED

一意に決めるのではなく、数名をリスト化する.

❖ 対象プロジェクト

2004.9.1 ~ 2005.5.31に報告されたのバグレポート.

➤ Eclipse

8655件のバグレポート.

➤ Firefox

9752件のバグレポート.

Step1. Characterizing Bug Report

“バグが過去のどのバグに似ているかを判断する”

❖ 存在するBugReportの特徴抽出

free-form text → 特徴ベクトル

❖ 特徴を用いて, 過去のreportsの特徴と比較

Step2. Labeling Bug Reports

”人出でバグにラベルをつける”

- ❖ assign-toは値として不適
- ❖ Heuristics to label the reports
バグを目視して、正しいLabelingを行う.
- ❖ 4つのルール
 - FirefoxにおいてFIXEDになっているもの
→最後にパッチを当てた人
 - EclipseにおいてFIXEDになっているもの
→reportをresolvedにした人
 - DUPPLICATEになっているreport
→DUPPLICATE
 - WORKSFORMEになっているreport
→クラス化不可能

Step3. Selecting Training Report

“機械学習をするための開発者を選ぶ”

- ❖ 活動しないDeveloperに割り当てても意味が無い
- ❖ 3ヶ月に9件以上のDeveloperを選択 ;;

Table 2: The effect of developer profile filtering on recommender accuracy and recall.

		# Dev.		Precision/Recall (%)	
		Firefox	Eclipse	Firefox	Eclipse
No Profile		414	146	23/1	58/7
>1 Fix in 3 mo.		94	82	59/2	57/7
Avg. Fixes Per Month Over 3 mo.	1	66	50	64/2	57/7
	2	33	42	59/2	57/7
	3	26	40	64/2	57/7
	4	21	40	64/2	58/7
	5	18	39	59/2	59/7
	6	13	37	45/1	57/7

Step4. Applying a Machine Learning Algorithm

“機械学習を適応する”

❖ NaïveBays

- 確率に基づく分類アルゴリズム.
- 実装が簡単、しかも高速.
- ベースラインとしてよく用いられる

❖ SVM

- 非線形的分類アルゴリズム.
- 未学習データに対して高い識別性能を得る.

❖ C4.5

- 決定木を使ったアルゴリズム.
 - 人間的で人が木を走査するのに似ている. ;;
-

Step4. Applying a Machine Learning Algorithm

Table 3: The effect of different machine learning algorithms on recommender accuracy and recall.

Predictions	Naïve Bayes		SVM		C4.5	
	Firefox	Eclipse	Firefox	Eclipse	Firefox	Eclipse
1	59/2	54/6	64/2	58/7	64/2	40/5
2	59/2	49/11	52/3	52/13	41/3	34/9
3	59/2	44/15	57/6	47/16	42/5	31/12

3つを試した結果, SVMを採用

Result

Table 2: The effect of developer profile filtering on recommender accuracy and recall.

		# Dev.		Precision/Recall (%)	
		Firefox	Eclipse	Firefox	Eclipse
No Profile		414	146	23/1	58/7
>1 Fix in 3 mo.		94	82	59/2	57/7
Avg. Fixes Per Month Over 3 mo.	1	66	50	64/2	57/7
	2	33	42	59/2	57/7
	3	26	40	64/2	57/7
	4	21	40	64/2	58/7
	5	18	39	59/2	59/7
	6	13	37	45/1	57/7

VALIDATING WITH GCC

GCCプロジェクトで検証を行う.

- ❖ 対象データ
194reports(2004.9-2005.3月)
- ❖ 結果
1つでPrecision Rateが6%, 複数で18%.
- ❖ 原因
 - プロジェクトの特徴が合わなかった.
 - 真の正確さを図ることが難しい.

検証は失敗

DISCUSSION -the value of this research

- ❖ 50%のPrecisionは高いといえるのか.
 - Eclipseでバグが再割当てされる割合 24%
⇒ Precision = 76%
 - 完全な自動化が目的ではない.
 - トリアージの時間短縮や正確さ向上
- ❖ なぜRecallの値が低くなってしまうのか.
 - PrecisionとRecallはトレードオフ
 - 計算方法が厳しく設定してある.
(正解集合が非常に大きい)

Summary

- ❖ 機械学習を用いたバグトリージの提示せ
 - 機械学習アルゴリズムにはSVNを使用
 - PrecisionとRecall
- ❖ 正確さの問題
 - 50～64%のPrecisionを達成.
 - 完全な自動化は出来ないが, 一助になる.
- ❖ 今後
 - その他の情報源も調査の対象に.
 - 適応可能な時期を決めること. ;;

参考文献

- ❖ SVMを使うと何が嬉しいの？
http://www.neuro.sfc.keio.ac.jp/~masato/study/SVM/SVM_1.html
 - ❖ データマイニングで使われるTop10アルゴリズム
<http://d.hatena.ne.jp/isseing333/20111123/1322053940>
 - ❖ SVMを使うと何が嬉しいの？
http://www.neuro.sfc.keio.ac.jp/~masato/study/SVM/SVM_1.html
 - ❖ 検索における適合率 (Precision) と再現率 (Recall)
<http://d.hatena.ne.jp/Zellij/20120214/p1>
-