

即时软件缺陷预测研究进展

蔡亮¹, 范元瑞¹, 鄢萌^{1*}, 夏鑫²



¹浙江大学, 杭州, 中国

²Monash大学, 墨尔本, 澳大利亚



MONASH University

汇报内容

1

即时缺陷预测简介

2

数据标记、特征提取和建模

3

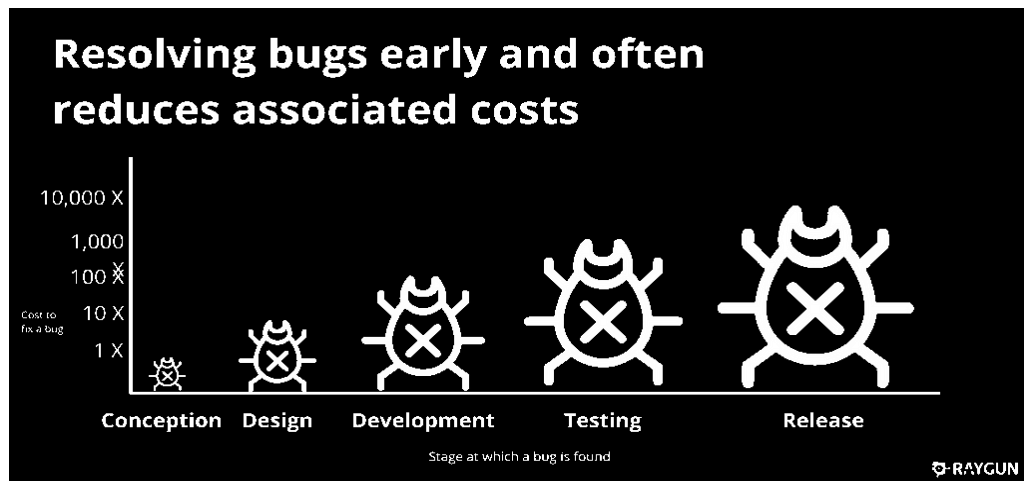
技术评估方法

4

关键问题与未来展望

缺陷预测 (Defect Prediction)

❑ 缺陷预测的动机



及时发现缺陷，辅助软件维护中的决策

❑ 目标与思路

目标：提前预测可能存在缺陷的软件实体（模块、文件或变更）

思路：利用机器学习技术，提取特征，构建分类/排序模型

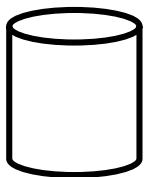
优化资源分配，节省维护成本

及时发现缺陷，提高代码质量

即时缺陷预测 (Just-in-Time Defect Prediction)

❑ 传统缺陷预测

预测粒度



模块、文件、类

存在问题

粗粒度

上下文难于追溯

❑ 即时缺陷预测

预测粒度



软件变更

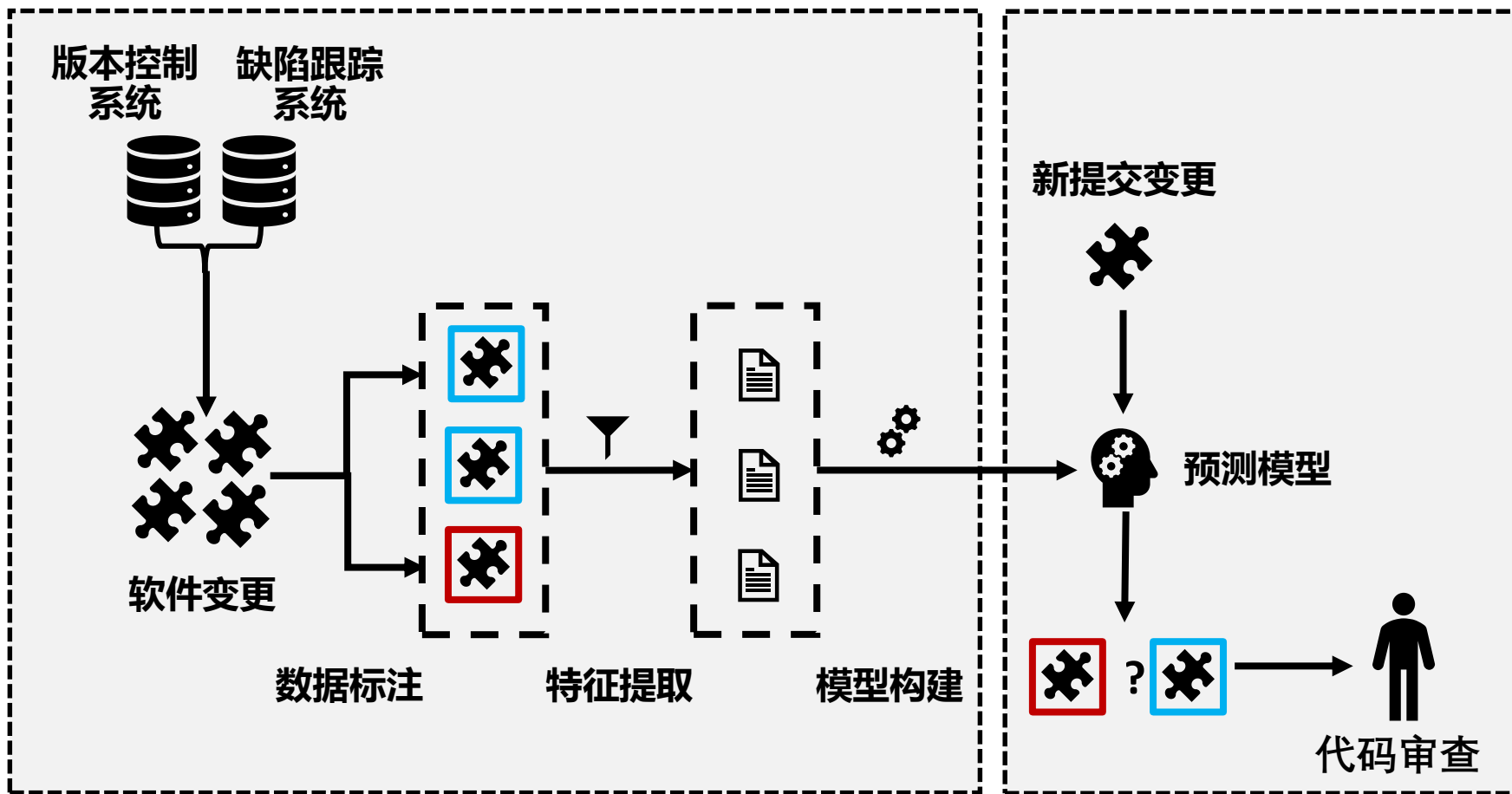
带来益处

细粒度

即时性

可追溯

即时缺陷预测一般框架



汇报内容

1

即时缺陷预测简介

2

数据标记、特征提取和建模

3

技术评估方法

4

关键问题与未来展望

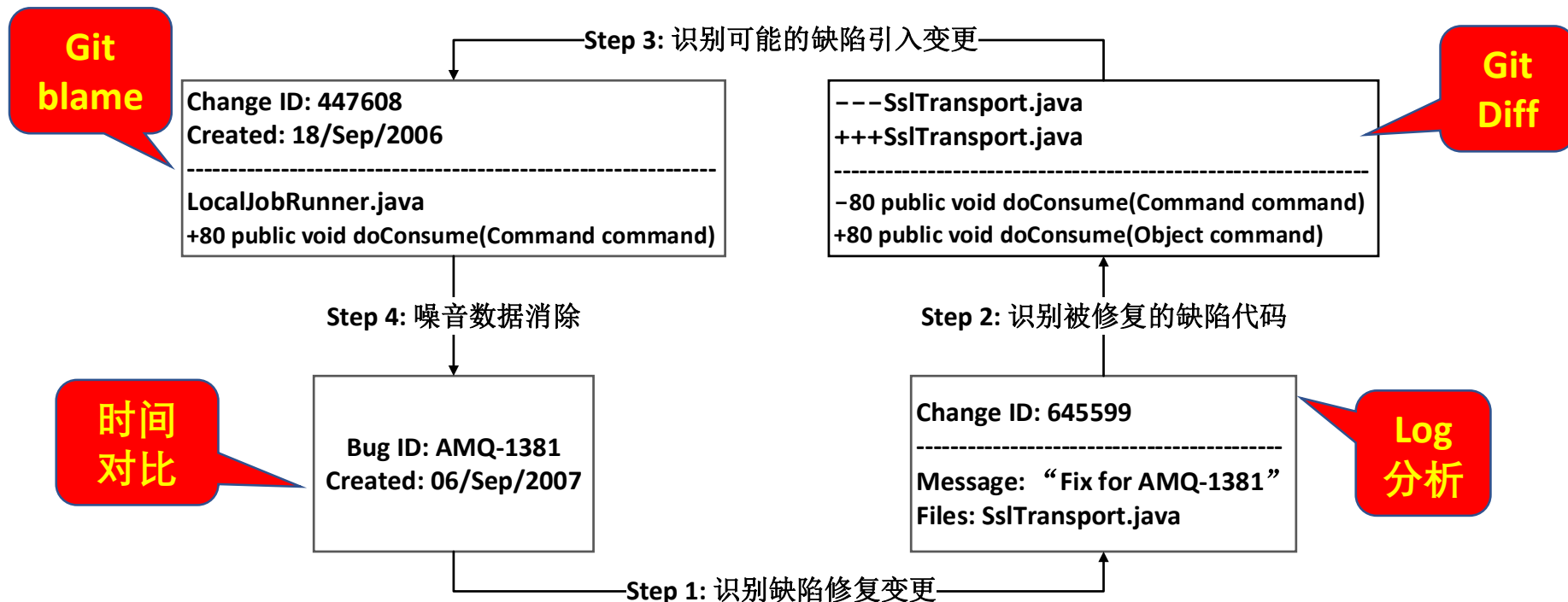
数据标注

□ SZZ算法: 识别缺陷引入变更的方法

由Sliwerski, Zimmermann, Zeller提出的一种**通用方法** (MSR'05)

*The SZZ algorithm is a **game changer**, because it provided a **new data source** for defect prediction studies. (Kamei and Shihab SANER' 16)*

□ SZZ算法的一般框架



SZZ算法改进与发展

代码演进图 SZZ
AG-SZZ (Annotation
Graph SZZ):

处理注释行、空行以及代码风格修改引入噪音

Kim et al. ASE' 06

元变更感知 SZZ
MA-SZZ (Meta-change
Aware SZZ)

处理Branch/Merge
Changes

Da Costa et al. TSE' 17

重构感知 SZZ
RA-SZZ (Refactoring
Aware SZZ)

在MA-SZZ基础上
处理重构类变更

Neto et al. Saner' 17

空行与代码风格噪音例子

```
public void notify(){  
-  
+ if (reportReferenceInfo){  
+     notifyAllUnknownReferences();  
+ }  
}
```

Change #1 (the real bug-introducing change)
if (x == null) return x;

Change #2 (format modification)
if (x == null)
 return x;

Change #3 (bug-fix)
if (x != null)
 return x;

特征提取



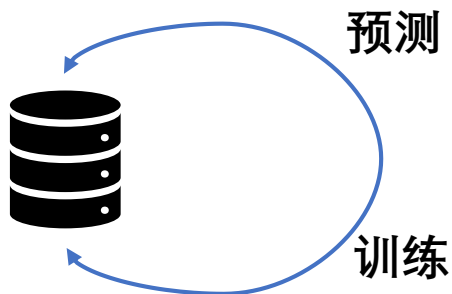
Meng Yan, et al. "Automating Change-level Self-admitted Technical Debt Determination". IEEE Transactions on Software Engineering (TSE), Accepted. 2018

模型构建

□ 有监督即时缺陷预测

利用有监督机器学习方法，从**已知标签**的代码变更数据中训练预测模型

同项目即时缺陷预测



✓ 需要足量历史数据

😊 较高的性能

✗ 新项目

跨项目即时缺陷预测

Kamei et al.
EMSE'16



✓ 新项目

😞 性能劣势

⚠ 需要选择合适的训练集或采用 Ensemble Learning 的方式

模型构建

□ 无监督即时缺陷预测

主要思想：直接利用变更特征值进行排序，达到effort-aware的目的。FSE'16

Effort-aware (工作量感知)：开发者根据变更排序来判定工作量，找到更多的缺陷

Effort (工作量或成本)：变更修正成本 (FLD)

例如：

EXP：变更修正文件数；

NE：变更修正文件数；

变更修正文件的...

有监督 → 无监督 → 有监督+无监督
分类 → 排序 → 分类+排序

建模区别：

考虑工作量感知应用场景

结果以排序方式呈现

Huang et al. ICSME'17：初始误报率问题；查准率、召回率

Fu et al. FSE'17：特征选择；

Huang et al. EMSE'18：有监督+无监督；

汇报内容

1

即时缺陷预测简介

2

数据标记、特征提取和建模

3

技术评估方法

4

关键问题与未来展望

验证方法

□ 基于ML的交叉验证方法

10次10折交叉验证



随机分层划分



能否模拟工具的使用场景？



□ 时间感知的验证方法 (Time-aware)

训练集

测试集



按时间窗口划分

Time

评估指标

基于IR的评估指标

查准率 (Precision)

查全率 (Recall)

F1-measure

正确率 (Accuracy)

AUC

隐含假设：以测试集总修改代码行数为
作为审查工作量预算

工作量感知 (Effort Aware) 的评估指标

隐含假设：以测试集总修改代码行数为
作为审查工作量预算

查准率 (Precision)

查全率 (Recall)

F1@20%

变更审查率 (PCI@20%)

初始误报数 (IFA)

一般机器学习设定 → 符合实际应用场景设定

汇报内容

1

即时缺陷预测简介

2

标记方法、特征提取和建模

3

技术评估方法

4

关键问题与未来展望

关键问题

□ 科学问题

可解释性

缺陷是什么？缺陷在哪里？

可操作性

如何修复该缺陷？

□ 技术难点

数据标注：

- ▲ Bug数据噪音；
- ▲ Diff数据噪音；

特征提取：

- ▲ 准确性: History需考虑Branch；
- ▲ 多样性: 多源异构与关联处理；

模型构建：

- ▲ 有监督与无监督结合；
- ▲ 前沿建模技术选择；

□ 工程实践

缺乏准确的
数据环境

缺乏多维的
数据特征

缺乏统一的
评估方法

未来展望

加强噪音数据的处理，改进数据标注方法

*Bug报告与Change映射问题；
SZZ中噪音代码过滤；*

综合考虑多源软件制品，提取多维特征

多源软件制品关联与特征提取；

研究先进的建模技术，取得建模技术上的突破

有监督+无监督；深度学习与集成学习；

实践应用场景中存在的问题

调研开发者对即时缺陷预测的看法；

推动即时缺陷预测的广泛应用



Question and Advice?

<http://yanmeng.github.io>

mengy@zju.edu.cn