

(动态比例的测试用例排序选择的持续集成优化)

Optimizing Continuous Integration by Dynamic Test Selection

张震宇

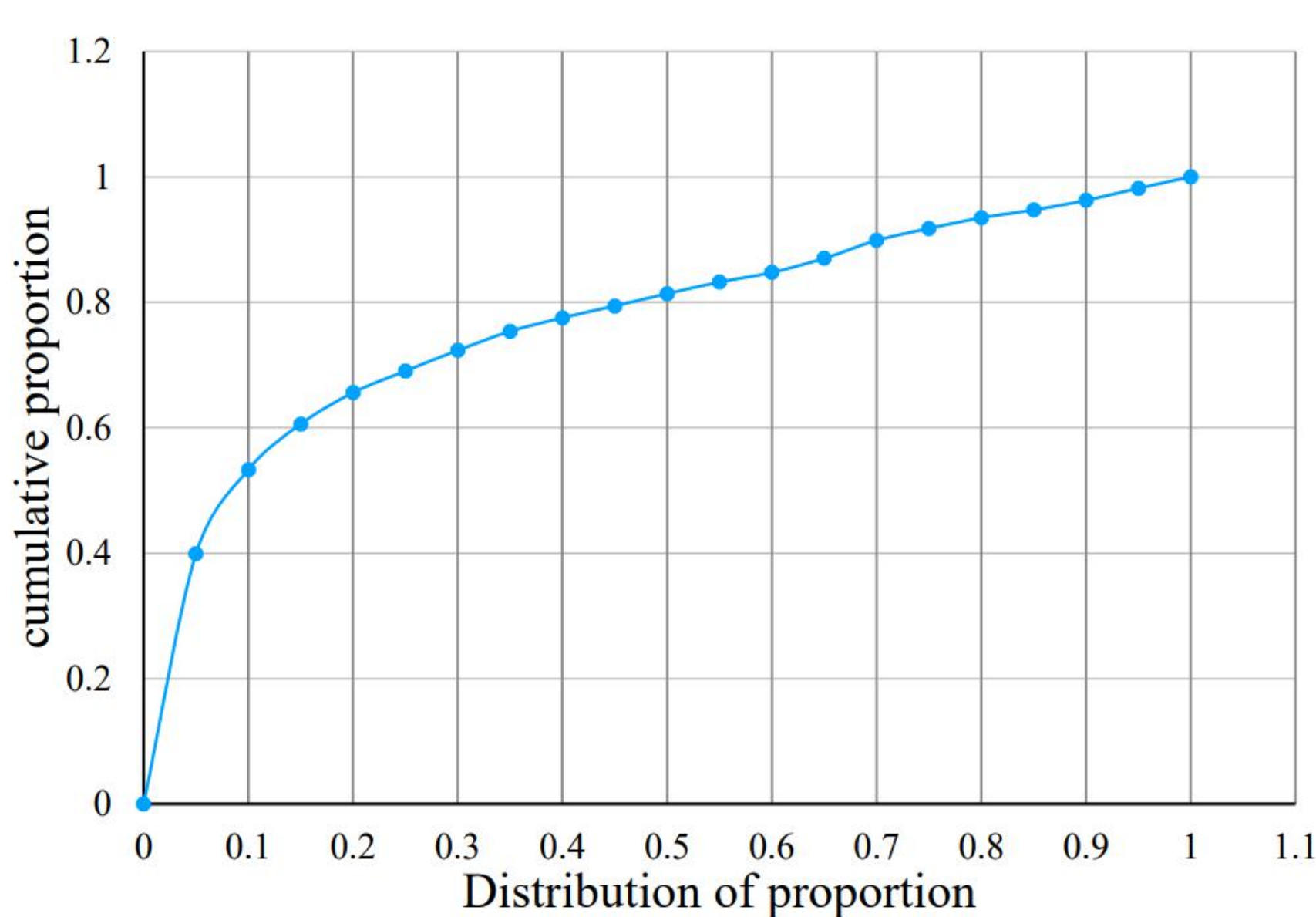
联系方式: 崔炳轶 cuiby@ios.ac.cn 13126897286

简介

- 持续集成是一种现代的软件工程实践方式,鼓励工程师频繁地进行代码提交和整合工作,以便尽早发现代码整合中可能出现的错误。虽然持续集成已经广泛应用于各种软件项目中,提高了软件开发的质量和效率,但是,持续集成的高昂开销成为软件开发效率的重要瓶颈。如何节省持续集成的构建开销,受到了广泛的关注。
- 在持续集成中,每个构建是一个完整的集成周期。节省持续集成构建开销的方法,目前分为两类。第一类是节省构建内的开销,第二类是跳过构建。第一类方法的代表性技术是测试用例排序选择。但是,该技术没有利用构建信息,缺乏构建区分度,对不同的构建使用相同的选择比例。第二类方法的代表性技术是跳过机器学习预测结果为“构建成功”的构建。但是该技术会跳过构建中的所有测试用例,当机器学习预测结果为假阴性结果时,会遗漏失败测试用例。

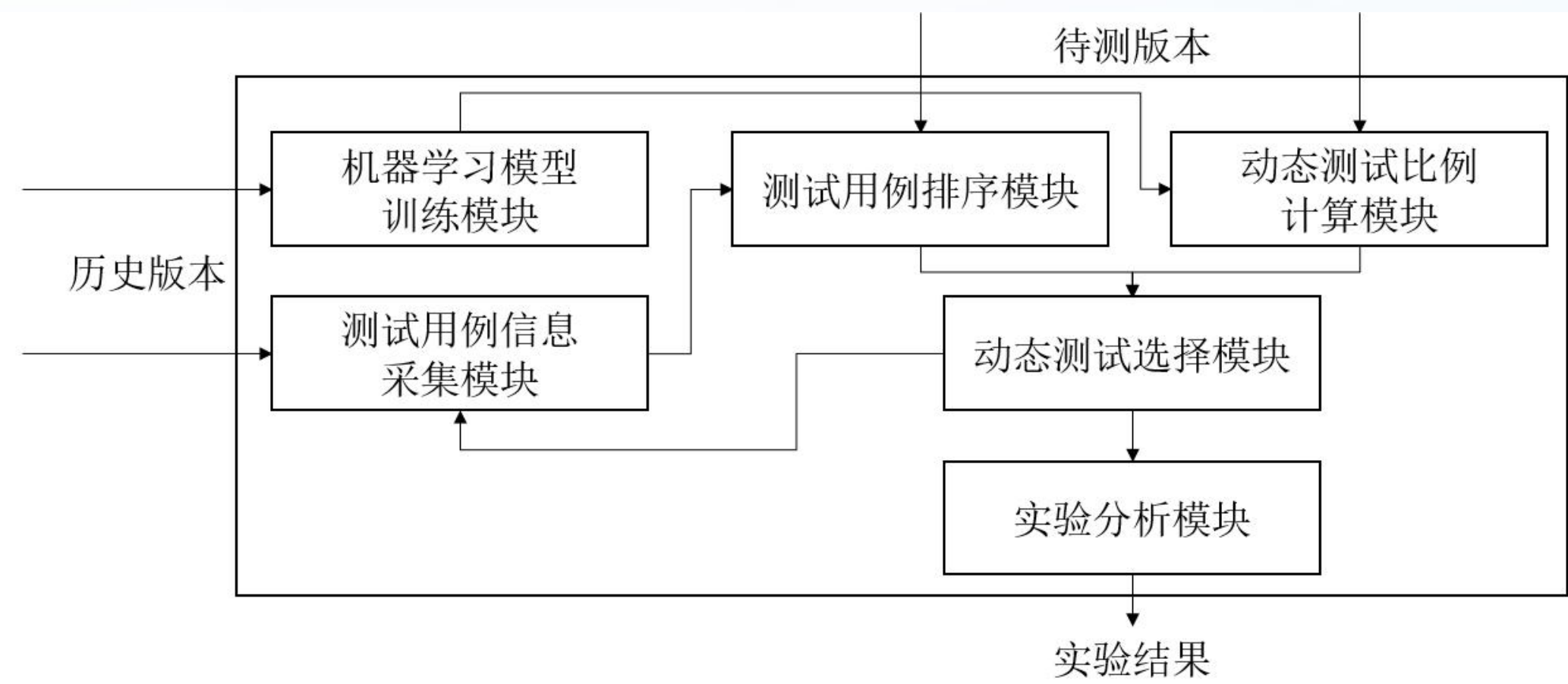
核心思路

- 提出了动态比例的测试用例排序选择技术,使用机器学习方法,对特定构建失败的概率以及必要测试比例进行预测,对于不同构建提供不同的测试用例选择比例。
- 必要测试比例(NTP)在TravisTorrent数据集分布



模型流程

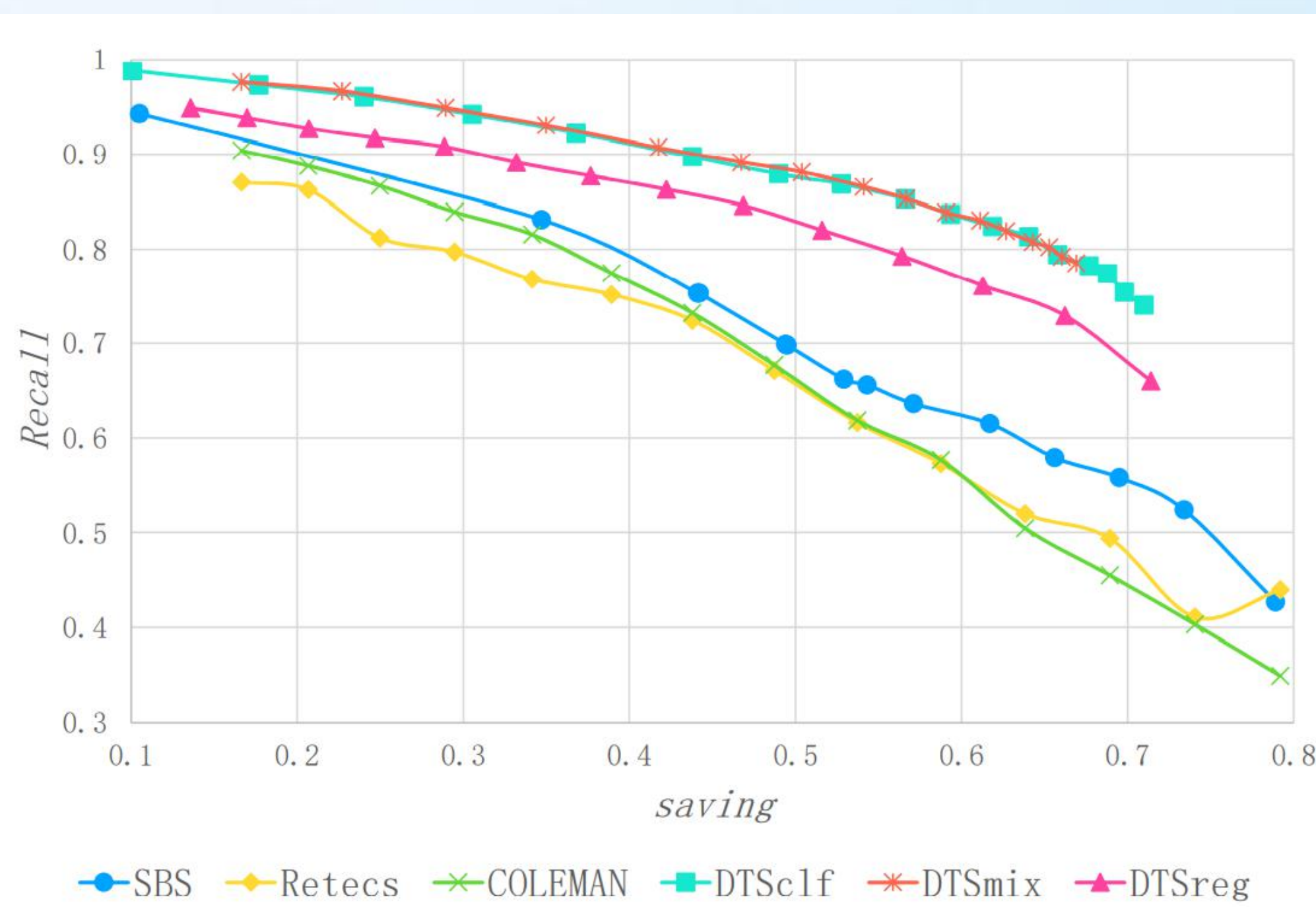
- 具体的算法流程概述如下:整体分为两个阶段,第一个阶段是训练阶段,DTS从构建中提取其特征,构成训练并收集各个项目的测试用例历史执行结果。在训练阶段结束时,DTS使用训练集训练机器学习模型。
- 第二个步骤是工作阶段。在此阶段中,对于每个构建,DTS使用训练得到的模型,提供一个测试用例选择比例;然后使用测试用例排序技术,生成一个排序的测试用例列表,然后按照优先级顺序,在选择比例内执行测试用例。



实验结果

- 在相似的总体开销(saving)下,DTS检测失败测试用例的能力(recall)明显高于三种现有技术(Retecs, COLEMAN和SBS)。
- DTS的fail-delay明显小于作为对照的Retecs, COLEMAN和SBS方法。

Method	saving	recall	efficiency	fail-delay
DTS _{clf}	0.490	0.879	0.629	0.88
DTS _{reg}	0.487	0.837	0.616	0.80
DTS _{mix}	0.487	0.889	0.629	1.38
Retecs	0.487	0.671	0.566	2.67
COLEMAN	0.487	0.677	0.567	15.47
SBS	0.495	0.698	0.579	1.80



- 对于不同项目的适用性,DTS的结果均优于三种现有技术

