Разработка системы для предсказания потенциально проблемных фрагментов программного кода

Студент – Коваленко Владимир Владимирович

Руководитель – Альперович Галина Сергеевна

СПбАУ РАН, 2015 г.

Метрики и баги

- Плотность багов сильно коррелирует с различными метриками кода и истории изменений
- ...по крайней мере, в зрелых проектах
- Часто качество кода измеряется подобными метриками

Метрики

- Метрики истории изменений
  - Характеристики изменений: частота, размер
  - Число авторов
  - Возраст файла
  - Время с последней правки
  - ...
- Метрики кода
  - Число строк, методов, классов
  - Цикломатическая сложность
  - Объектно-ориентированные метрики
  - Метрики связности кода
  - ...
- Другие метрики
  - Баг-трекер
  - История запуска тестов
  - ...

Общий подход

- Сбор данных о проекте
- Подсчёт метрик
- Анализ
- Использование результатов

Возможные подходы к анализу:

- Значимая метрика
- Композитная метрика
- Машинное обучение

Общие выводы теоретических работ

- SL-модели для Bug Prediction в целом работают
- Универсальной модели нет: проекты разнородные
- Модели с метриками кода работают лучше, чем без них
- Точность/полнота 0.7
- Почти все модели проверялись на ограниченном наборе проектов

Отдельные упоминания об использовании в индустрии:

- Microsoft Research
- IBM

Google Bug Prediction Score

Формула

$$score = \sum_{i=0}^{n} \frac{1}{1 + e^{-12t_i+12}}$$

Сумирование по bugfix-коммитам, затрагивающим данный файл
bugfix определяется ключевыми словами в commit message t_i - относительная дата i-го коммита (от 0 до 1)

«Классический» результат, один из эталонов для оценки качества

Цель и задачи

Цель:

- Рабочее решение для предсказания опасных файлов
  - Пригодное для человеческого использования
  - Модель не привязана к платформе
  - **Не требует человеческого обучения**

Задачи:

- Оценка качества модели: методика
  - Для подбора оптимальных параметров
- Генерация суррогатных обучающих данных
- Модель
  - Структура модели: формула или ML?
  - Индекс
  - Не зависит от платформы
- Окружение для модели
  - Сбор данных
  - Подсчёт метрик
  - UI

Оценка качества: метод

- Обучающие данные неточны
- В идеале – человеческая оценка (всё-таки это тул)
- Точных тестовых данных просто нет
- Оцениваем через баг-трекер

Генерация обучающих данных

- Разработан инструмент автоматической классификации коммитов на bugfix- и не-
- Точность и полнота около 0.7
- Инструмент может быть пригоден для сравнения моделей **между собой**, но не для абсолютной оценки качества

Prediction Model

Не привязанная к TeamCity модель

- Machine Learning by Weka
- Легко переносится на любую платформу
- Дополнительные особенности
  - Информация о ревизиях может передаваться в произвольном порядке
  - При добавлении новой ревизии пересчитываются устаревшие метрики

Используемые в модели метрики

- Число авторов
- Частота правок
- Возраст файла
- Число затрагивающих файл коммитов
- Google Score
- Тип связанных Issue

Model Host: TeamCity plugin

- Плагин для TeamCity, собирающий необходимые для модели данные
- Данные:
  - VCS
  - Issue Tracker
  - Также поддерживаются другие источники данных
    - Статический анализ (e.g. Sonar)
    - История сборок (тесты)

Модель

- Классификатор: Naive Bayes / Decision Tree
- Опасными считаем топ X файлов по выводу классификатора
- Оцениваем через баг-трекер

Пример результатов оценки качества

Данные для репозитория Kotlin за последние 2 года.
Вероятность попадания файла в bugfix-коммит:

- наугад взятого файла – 0.42
- после любой правки – 0.50
- после попадания в top-5 по Google Score – 0.63
- после попадания в top-5 по выводу Naive Bayes – 0.87, $\sigma = 0.05$
- после попадания в top-5 по выводу Decision Tree – 0.88, $\sigma = 0.05$

Выводы

- Разработана система предсказания опасных файлов:
  - Окружение для модели
  - Модель для предсказания опасных файлов
  - Инструмент для генерации обучающих данных
- Каждый компонент может быть при необходимости переиспользован независимо
- Разработаны методика и инструмент для оценки относительного качества выдачи модели без участия человека
- Показана целесообразность применения машинного обучения для подобных задач

Спасибо!

- JetBrains
  - Галине Щёкотовой
  - Евгению Кошкину
  - Евгению Пасынкову
  - Леониду Хачатурову
  - Максиму Подколзину
  - Сергею Шкредову
- Семёну Прошеву

Вопросы?
https://github.com/vovak/tc-bp-plugin