Тестовое задание на проект
Logical coupling analysis in Git repositories

Roman Kovalev

Май 2024

Введение

Тестовым заданием на проект является разработка инструмента для расчета пар разработчиков, которые чаще всего контрибьютят в одни и те же файлы/модули репозитория GitHub. В данном отчете описано что было реализовано при работе над тестовым заданием, однако основная его ценность - описание того, что и как еще можно сделать в дальнейшем.

Анализ задачи

Итак, у нас есть GitHub репозиторий. Это означает что у нас есть список коммитов в репозиторий со всей содержащейся в них информацией: дата коммита, автор и измененные файлы.

Для начала отметим, что имеет смысл ограничить количество коммитов, которые мы будем анализировать. Это необходимо, поскольку сама идея "разделить разработчиков по похожести"подразумевает актуальные данные и по множеству действующих разработчиков, и по интересам текущих участников проекта. В дополнение стоит сказать что репозитории крупных проектов включают в себя десятки и сотни ежедневных коммитов, что может дать нам массу нерелевантной информации. Поскольку нас интересует актуальная информация, то разумно будет ограничиться коммитами за несколько последних дней.

Теперь, когда мы определились со входными данными, следует ввести численную оценку "похожести"двух разработчиков по используемым ими файлам/модулям. Это можно сделать несколькими способами, но мы выберем самый простой: количество файлов, куда вносили изменения и первый и второй разработчики.

Теперь можно рассмотреть полный взвешенный граф, вершины которого отвечают разработчикам, а веса ребер - количеству общих файлов для соответствующей пары.

Тогда исходную задачу можно перефразировать следующим образом: *Найти паросочетание максимального веса в полном взвешенном графе без отрицательных весов*.

Реализованное решение

В своем решении я реализовал жадный алгоритм, на каждой итерации выделяющий пару двух максимально похожих среди всех свободных разработчиков.

Асимптотика такого алгоритма $O(n^3)$.

К сожалению, жадный алгоритм не дает гарантии что найденное паросочетание будет иметь максимальный вес, но некоторое псевдооптимальное разбиение мы получим.

Точное решение

Давайте сведем нашу задачу к Задаче о назначениях.

Задача о назначениях:
Дан двудольный взвешенный полный граф. Требуется найти паросочетание максимального веса.

Пусть исходный граф представляется так: $G = \{V,\ E = V \times V,\ f : E \to \mathbb{Z}_+\}$ Давайте продублируем каждую вершину - множество V' - и рассмотрим граф H:

$$H = \left\{V \cup V',\ E',\ g : E' \to \mathbb{Z}_+\right\}$$

Вес ребра в графе H между двумя вершинами равен нулю, если они обе принадлежат одному множеству V или V', и равен весу ребра между их праобразами в исходном графе в ином случае. Вес ребра между копиями одной вершины также равен нулю.

Теперь мы имеем двудольный взвешенный полный граф $H_{n,n}$ вершинах. Если мы найдем решение задачи о назначениях на текущем графе, то очевидным образом восстановим паросочетание максимального веса на исходном графе.

Стоит упомянуть что задача о назначениях решается за $O(n^4)$. Сведение исходной задачи к задаче о назначениях работает за $O(n^2)$, поэтому наилучшего разделения на пары можно добиться за $O(n^4)$.

В статье показано, что у задачи о назначениях есть рандомизированные решения с худшей асимптотикой, но лучше работающие на практике. В нашем случае будет разумно использовать одно из таких решений.

Демонстрация алгоритма

За тестовый репозиторий был взят публичный репозиторий библиотеки pytorch.

1. Запускаем файл *Main.py*.
2. Далее следует указать полный путь до склонированного репозитория (*/home/roman-not-hehe/Desktop/sandboxes/pytorch*)
3. Указываем количество дней за которые будем анализировать репозиторий.
4. Получаем информацию о количестве коммитов авторов, вносивших изменения в репозиторий за это время и общее количество коммитов.
5. При разделении на пары почти наверняка получатся "остаточные"пары разработчиков, то есть такие, что их "похожесть"равна 0. Поэтому мы должны выбрать: выводить эти пары на экран или нет.
6. Далее на экран выводятся пары разработчиков с числом файлов, в которые они оба вносили изменения.
7. В самом конце программа выводит среднее количество общих файлов для пар, а так же их медианное значение.

The first example

The second example