

Глава 1

Алгоритмы

1.1 Итеративные алгоритмы

Цепь Маркова Строится матрица переходных вероятностей с учетом коэффициента d . Затем на ней итеративно вычисляется стационарный вектор с заданной точностью, который затем возвращается.

Power Method Для каждой страницы начальным рангом назначается единица, а затем выполняется итеративный пересчет рангов. Пересчет останавливается, когда разница между предыдущим вектором рангов и текущим станет меньше заданной.

Adaptive Power Method Улучшение предыдущего метода. Пересчет останавливается для конкретных компонент вектора рангов, когда разница становится меньше заданной точности.

Extrapotaling Adaptive Power Method Улучшение предыдущего метода. Каждые n шагов алгоритм высчитывает уточняющее значение исходя из предыдущий шагов алгоритма, которое впоследствии используется для уточнения текущего значения.

1.2 Алгоритмы Монте Карло

Endpoint Random start Monte Carlo Method Метод выполняет n или n^2 симуляций проходов, завершающихся с вероятностью $(1 - d)$, начиная со случайной страницы, заданное число раз. Конечная страница увеличивает показатель ранга на 1. Возвращает итоговый вектор сумм.

Endpoint Cyclic Start Monte Carlo Method Метод выполняет заданное число симуляций проходов для каждой страницы, завершающихся с вероятностью $(1 - d)$. Конечная страница увеличивает показатель ранга на 1. Возвращает итоговый вектор сумм.

Complete Path Monte Carlo Method Метод выполняет заданное число симуляций проходов для каждой страницы, завершающихся с вероятностью $(1 - d)$. Все страницы, посещенные в процессе прохода, увеличивают показатель ранга на 1. Возвращает итоговый вектор сумм.

Stopping Complete Path Monte Carlo Method Метод выполняет заданное число симуляций проходов для каждой страницы, завершающихся с вероятностью $(1 - d)$ или при попадании на страницу без внешних ссылок. Все страницы, посещенные в процессе прохода, увеличивают показатель ранга на 1. Возвращает итоговый вектор сумм.

Random start Stopping Complete Path Monte Carlo Method Метод выполняет n или n^2 симуляций проходов, завершающихся с вероятностью $(1 - d)$, начиная со случайной страницы, заданное число раз. Все страницы, посещенные в процессе прохода, увеличивают показатель ранга на 1. Возвращает итоговый вектор сумм.

Глава 2

Тесты

2.1 Описание тестов

Все тесты (кроме Level Test) используют заранее подготовленные файлы с известными ответами. Первый и третий файл - примеры, найденные в интернете. Второй был сгенерирован с помощью SeparatedLevelsTreeGen, результат на нем просчитан с помощью Power Method.

Position Test Тест создает списки страниц, упорядоченные по убыванию ранга, для истинного и полученного результатов. После этого считает количество совпадений элементов, расположенных на местах с одинаковым номером и возвращает процент совпадений.

Sequence Test Тест создает списки страниц, упорядоченные по убыванию ранга, для истинного и полученного результатов. После этого считает количество совпадений элементов, расположенных на местах с одинаковым номером. Если номера страниц не совпадают, то страница, стоящая в списке полученного результата, удаляется. Это исключает ситуацию, когда мы дважды учитываем ошибку места страницы.

Vector Test Тест нормализует истинный и полученный списки рангов, после чего возвращает расстояние между векторами по метрике l_1 .

Distance Test Тест высчитывает расстояние каждой страницы от её истинного положения, после чего возвращает средний результат.

Level Test Тест использует `BinaryTreeGenerator`, для генерации графа, который являлся бы бинарным деревом. Для данного дерева нам наперёд известны позиции страниц в ранговом списке. Тест возвращает процент страниц с правильно определённым уровнем.

Top Test Тест считает процент совпадений первых i страниц, для всех i меньших заданного числа, и возвращает список процентов совпадений для каждого i .

2.2 Запуск тестов

Чтобы запустить проводимые мною тесты, достаточно запустить метод `CompleteTest()` из модуля `Test`, то есть выполнить следующий код:

```
# Python
import PageRank.Test as t
t.CompleteTest()
```

Глава 3

Результаты алгоритмов на тестах

3.1 Итеративные алгоритмы

Все итеративные алгоритмы выдают полностью правильные ответы на примерах (первый и третий тесты), при этом на втором тесте в ответах появляются расхождения из-за специфики работы тех или иных алгоритмов, но при этом лидирующие страницы определяются точно. Исключением является цепь Маркова, она имеет серьезные расхождения с Power Method и его улучшениями.

3.2 Алгоритмы Монте Карло

Endpoint Random Algorithms Алгоритмы данной группы показывают неплохие результаты на примерах, на втором тесте показатели хуже, но топ страниц определен достаточно точно, при этом они отработали примерно в 6 раз быстрее Extrapolating Adaptive Power Method.

Complete Path Algorithms Алгоритмы показывают не очень хорошие значения части тестов, что частично связано с малым заданным числом проходов (алгоритмы в тестах настроены на небольшое число страниц). Имеют преимущества предыдущей группы алгоритмов: скорость, высокое качество определения лидирующих страниц, но имеют лучшие значения тестов. В этой группе выделяется Stopping Complete Path Monte Carlo Method, так как все страницы без ссылок на них имеют одинаковое значение ранга.