МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ   
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ   
«ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт математики и информационных систем

ФАКУЛЬТЕТ КОМПЬЮТЕРНЫХ И ФИЗИКО-МАТЕМАТИЧЕСКИХ НАУК

КАФЕДРА ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ

Допущен к защите

Заведующий кафедрой ПМИ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е.В. Разова

**Разработка системы ссылочного ранжирования**

Курсовой проект по дисциплине  
«Теория алгоритмов»

Выполнил студент группы ФИб-3301-51-00     / Д. О. Ощепков/

Руководитель к.т.н.,

доцент кафедры ПМИ     / А. Г. Татаринова/

Работа защищена с оценкой     \_\_\_\_.\_\_\_\_.2024 г.

Члены комиссии:     / /

    / /

Киров 2024

Содержание

[Введение 3](#_Toc166515284)

[1 Разбор алгоритмов ссылочного ранжирования 5](#_Toc166515285)

[1.1 Постановка задачи 5](#_Toc166515286)

[1.2 Метрики оценки качества 5](#_Toc166515287)

[1.2.1 Pairwise orderedness 6](#_Toc166515288)

[1.2.2 Precision 7](#_Toc166515289)

[1.2.3 Recall 7](#_Toc166515290)

[1.2.4 метрика и ее частный случай метрика 8](#_Toc166515291)

[1.3 Обзор рассматриваемых алгоритмов 8](#_Toc166515292)

[1.3.1 Pagerank 8](#_Toc166515293)

[1.3.2 HITS 11](#_Toc166515294)

[1.4 Выводы по разделу 12](#_Toc166515295)

[2 Применение алгоритмов на практике 13](#_Toc166515296)

[2.1 Какие-то данные 13](#_Toc166515297)

[2.2 Выводы по разделу 13](#_Toc166515298)

[Заключение 14](#_Toc166515299)

[Библиографический список 15](#_Toc166515300)

[Приложения 16](#_Toc166515301)

[Приложение А. Листинг программы 16](#_Toc166515302)

# Введение

Надо начать раньше, например в целом про ранжирование, затем перейти к ссылочному ранжированию и графам

Основная идея алгоритмов ссылочного ранжирования заключается в том, чтобы определить степень важности каждого элемента на основе структуры графа, образованного самими элементами и связями между ними.

Эти алгоритмы часто применяются в поисковых системах для ранжирования результатов поиска (<https://papers.cumincad.org/data/works/att/2873.content.pdf> ), в социальных сетях для определения влиятельных пользователей или контента (<https://link.springer.com/article/10.1007/s13278-014-0172-x> ), в анализе сетей для выявления ключевых узлов (<https://www.mdpi.com/1996-1073/15/3/797> )

Основные алгоритмы ссылочного ранжирования:

Pagerank и его вариации, такие как personalized pagerank, weighted pagerank, personalized weighted pagerank. Они используют итеративный процесс для расчета весов страницы, учитывая количество и качество входящих ссылок. (<https://papers.cumincad.org/data/works/att/2873.content.pdf> )

Hits и его вариации. Используется для определения важности веб-страниц на основе их характеристик как "хабов" и "авторитетов". Хабы (Hubs) - это страницы, которые содержат много ссылок на другие страницы. Чем больше ссылок у страницы, тем она более "хабовая". Авторитеты (Authorities) - это страницы, на которые много ссылается. Такие страницы считаются важными и качественными и обладают высоким "авторитетом". (<https://dl.acm.org/doi/pdf/10.1145/324133.324140> )

Проблема исследования заключается в отсутствии четких рекомендаций или наставлений для выбора конкретного алгоритма ссылочного ранжирования классификации в конкретной задаче. Каждый из методов обладает своими сильными и слабыми сторонами, а также применим к определенным условиям и задачам.

Для решения данной проблемы необходимо провести сравнительный анализ различных алгоритмов ссылочного ранжирования в различных сценариях и задачах.

Для этого в рамках курсового проекта предлагается исследовать возможности алгоритмов ссылочного ранжирования при решении задачи поиска рекомендаций в зависимости от пользовательских предпочтений.

Это позволит определить, какие методы лучше справляются с определенными типами данных, объемом данных, структурой сети и характеристиками узлов. Кроме того, такой анализ может учитывать требования к точности ранжирования, скорости работы алгоритма

Целью курсового проекта является формирование рекомендации по выбору алгоритма ссылочного ранжирования.

Для достижения цели необходимо решить следующие задачи.

Обзор литературы. Изучить существующие научные статьи, книги и исследования, посвященные метрическим методам классификации. Выявить основные методы и их характеристики, а также рассмотреть примеры исследований, где эти методы применялись в различных задачах.

Описание и анализ методов ранжирования

Подготовка набора данных. Собрать или использовать доступные наборы данных, соответствующие различным предметным областям. Обеспечить разнообразие данных в терминах признаков, классов, объема выборки и сложности задачи.

Выбор метрик для сравнения алгоритмов.

Реализация и применение каждого из методов для решения задач ранжирования на различных наборах данных.

Анализ результатов и сравнение эффективности каждого метода на различных наборах данных. Определить подходящие метрики оценки качества классификации и провести серию экспериментов, сравнивая различные метрические методы классификации на подготовленных наборах данных. Сравнивать производительность методов в терминах точности, полноты, F-меры.

Сделать выводы о применимости каждого метода в различных сценариях и задачах ранжирования.

Курсовой проект включает два раздела. В первом разделе проведен обзор литературы и алгоритмов по предметной области. Во втором разделе был проведен анализ алгоритмов ранжирования.

# Разбор алгоритмов ссылочного ранжирования

## Постановка задачи

Дано:

* Множество объектов , где - количество объектов.
* Ориентированный граф , представляющий связи между объектами. Ребро означает, что объект "ссылается" на объект .

Алгоритм должен присвоить каждому объекту числовой ранг , отражающий его относительную "важность" в контексте графа .

## Метрики оценки качества

В дальнейшем нам понадобится следующее:

|  |  |  |  |
| --- | --- | --- | --- |
| Category | |  | |
| YES | NO |
|  | YES |  |  |
| NO |  |  |

Таблица 1.2.1. The Contingency Table for Category . [9, p. 33]

(Sebastiani F. Machine learning in automated text categorization //ACM computing surveys (CSUR). – 2002. – Т. 34. – №. 1. – С. 1-47.) Таблица адаптирована под курсовой проект.

Здесь – Истинно положительный (True Positive), – Ложно Положительный (False Positive), – Ложно Отрицательный (False Negative), – Истинно Отрицательный (True Negative).

Метрики ниже взяты из статьи (<https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=0bee052af002eb197277cd222d62154c7de4ac8a> , 578)

Формализуем человеческую оценку в функциональный вид

Где , а – множество вершин графа страниц

Хорошая страница или плохая можно определить по-разному на практике. Этой оценкой может быть, например, то, что значение, даваемое экспертом, выше, чем среднее значение по всем оценкам страниц.

### Pairwise orderedness

Для данной метрики понадобиться вспомогательная функция, которая будет сигнализировать, если плохая страница получила равный или более высокий рейтинг.

– функция, возвращающая ранг вершины согласно алгоритму, а – это вершины графа.

Теперь можем сформулировать pairwise orderedness

Если pairord равна 1, не существует случаев, когда T неверно оценил пару. Наоборот, если pairord равна нулю, то T неверно оценил все пары.

### Precision

Показывает долю высоко оцененных вершин, которые действительно являются хорошими, от всех высоко оцененных вершин.

Precision показывает способность модели точно предсказывать положительный класс. Если вы хотите избегать ложных срабатываний и быть уверенными в том, что предсказанный положительный класс действительно является положительным, вам нужно стремиться к высокому значению precision.

Формулу можно преобразовать в следующее:

### Recall

Показывает долю хороших вершин, которые высоко оценены алгоритмом, от всех хороших вершин.

Recall показывает способность модели обнаруживать все положительные примеры. Если важно, чтобы модель обнаруживала как можно больше положительных случаев, даже если это приводит к большему количеству ложных срабатываний, то стоит стремиться к высокому значению recall.

Формулу можно преобразовать в следующее.

### метрика и ее частный случай метрика

Является объединением Precision и Recall, показывает приоритет одной метрики над другой. При метрики учитываются в равной степени. [9, p. 33, 36]

-мера полезна если данные несбалансированны, так как она учитывает и точность, и полноту, что позволяет получить общую оценку производительности модели, не учитывая смещение в сторону преобладающего класса.

## Обзор рассматриваемых алгоритмов

### Pagerank

Входные данные:

* - матрица переходов графа, где - вероятность перехода от узла к узлу .
* - вектор персонализации, отражающий предпочтения пользователя к определенным страницам.
* - коэффициент, определяющий вес матрицы переходов.
* - коэффициент, определяющий вес вектора персонализации.
* - коэффициент, необходимый для чтобы каждая веб-страница имела самоцикл с ненулевой вероятностью, а также каждая пара веб-страниц была связана.
* - количество итераций.

Алгоритм:

1. Инициализация:  
   Создаем вектор PageRank и устанавливаем начальные значения:
2. Итеративный процесс:

Для от 1 до :

Выходные данные:

* вектор PageRank, где представляет рейтинг узла .

Замечания:

* Коэффициенты , и должны удовлетворять условию .
* Вектор персонализации должен быть нормализован, так что сумма его элементов равна 1.
* Количество итераций iterations влияет на точность результатов.

Использование разреженных матриц для представления графа значительно улучшает эффективность алгоритма PageRank, особенно для больших графов с небольшим количеством связей.

Временная сложность:

Цикл выполняется iterations раз.

Умножение разреженной матрицы на вектор: - поскольку мы перебираем только ненулевые элементы матрицы.

Умножение вектора на вектор: .

Сложение векторов:

Итого: Временная сложность алгоритма PageRank с использованием разреженных матриц составляет .

### HITS

Входные данные:

* : Разреженная матрица, представляющая ориентированный граф, где ненулевое, если есть ребро от узла к узлу .
* iterations: Число итераций алгоритма.

Алгоритм:

1. Инициализация:
   * Создать векторы hubs и authorities размера n (число узлов в графе).
2. Итеративный процесс:
   * Повторить iterations раз:
     + Обновление authorities:
       - Для каждого узла j:
     + Обновление hubs:
       - Для каждого узла j:
     + Нормализация:
       - Разделить все элементы hubs и authorities на их евклидову норму.

Где ­ мера авторитетности узла , — мера хабности узла . и ­ обновленные значения и соответственно. (<https://dspace.mit.edu/bitstream/handle/1721.1/86737/49317146-MIT.pdf?sequence=2&isAllowed=y> стр23)

Выходные данные:

* hubs: Вектор, содержащий hub оценки для каждого узла.
* authorities: Вектор, содержащий authority оценки для каждого узла.

Замечания:

* Использование разреженных матриц для представления графа значительно улучшает эффективность алгоритма HITS, особенно для больших графов с небольшим количеством связей.
* Нормализация предотвращает экспоненциальный рост значений hubs и authorities в процессе итераций.

Временная сложность:

Цикл выполняется iterations раз.

Умножение разреженной матрицы на вектор: - поскольку мы перебираем только ненулевые элементы матрицы.

Нормализация: – нахождение суммы вектора и деление каждого элемента на эту сумму.

Итого: Временная сложность алгоритма PageRank с использованием разреженных матриц составляет .

## Выводы по разделу

# Применение алгоритмов на практике

## Какие-то данные[[1]](#footnote-1)

## Выводы по разделу

# Заключение

# Библиографический список

# Приложения

## Приложение А. Листинг программы

1. [↑](#footnote-ref-1)