Санкт-Петербургский государственный университет

Кафедра технологии программирования

Филин Илья Владимирович

Выпускная квалификационная работа бакалавра

Построение характеристического вектора клик веб-графа и его применение для анализа структуры сайтов

Направление 010400

Прикладная математика, фундаментальная информатика и программирование

Заведующий кафедрой,
кандидат физико-математических наук,
доцент

Сергеев С. Л.

Научный руководитель,
доктор технических наук,
профессор Печников А. А.

Рецензент,
“ООО Искусство управления данными”,
старший программист Чернобровкин Д. И.

Санкт-Петербург

2016

Содержание

Введение 3

Постановка задачи 6

Глава 1. Характеристический вектор клик веб-графа 7

Глава 2. Программа сканирования сайта для построения его веб-графа 9

Глава 3. Программа поиска клик 11

Глава 4. Эксперименты с сайтами СПбГУ 16

4.1. Сканирование сайтов 16

4.2. Построение векторов клик веб-графа 19

Глава 5. Кластерный анализ 23

5.1. Общие сведения 23

5.2. Кластеризация результатов работы программы построения векторов клик веб-графа 26

Выводы 28

Заключение 29

Список литературы 30

Введение

Веб-граф[1] является достаточно распространенной моделью веб-пространства. Такая модель используется на протяжении последних 15-20 лет для описания веба и вебометрических исследований. Веб-граф очень удобен для отслеживания связей между различными веб-ресурсами.

Для его построения необходим список веб-сайтов, которые будут выступать в качестве множества вершин, и список гиперссылок между этими сайтами, которые будут составлять множество дуг веб-графа. Такой граф будет иметь ребра-петли, и он будет ориентированным.

Крупное исследование веба с применением веб-графа в 1999 году произвел Андрей Бредер вместе с коллегами[2]. Они создали математическую модель веб-пространства, и назвали ее “Bow tie” (см.рис.1).

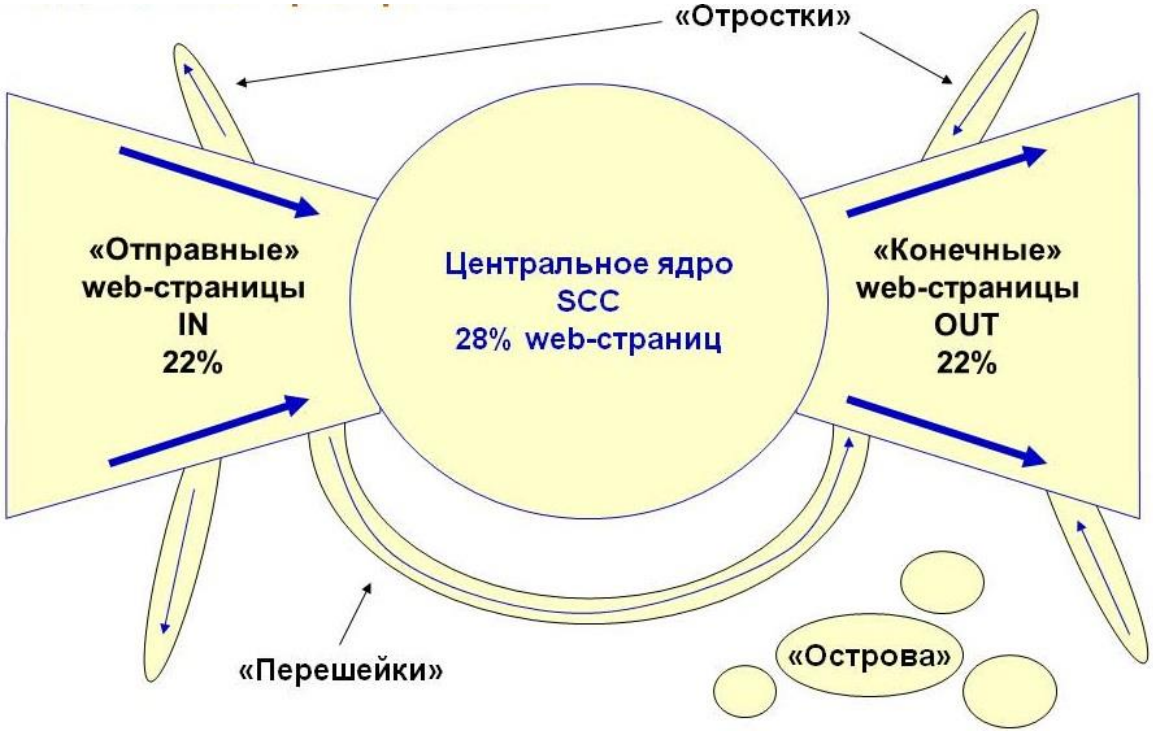

Рис.1. Модель веб-пространства “Bow tie”

Исследовав более 200млн. страниц ученые построили веб-граф и, проанализировав его, пришли к выводу, что все страницы можно разделить на 5 множеств:

1. Центральное ядро – тесно связанные страницы, то есть с любой страницы этого множества можно перейти на любую другую страницу этого множества по конечному числу гиперссылок.
2. “Отправные” web-страницы – на этих страницах содержатся гиперссылки, ведущие в центральное ядро, но из ядра к ним попасть невозможно.
3. “Конечные” web-страницы – страницы, обратные к отправным, к ним можно попасть из ядра, но в ядро с них попасть невозможно.
4. “Отростки”, “перешейки” – полностью изолированные от центрального ядра страницы.
5. “Острова” (или hidden web) – страницы, на которые не существует гиперссылок. Попасть на них можно только зная их адрес.

Однако, в современном мире количество веб-ресурсов становится все больше и больше, количество сайтов стремительно увеличивается, соответственно растет и количество гиперссылок между ними. Так же растет количество страниц, входящих в множество “островов”. Поэтому исследование веба в целом было выполнимо только на начальных этапах его развития, в наше же время исследования представляются возможными лишь на сравнительно небольших фрагментах веб-пространства.

В качестве таких фрагментов можно рассматривать веб-сайты[2]. Эти фрагменты так же хорошо описываются веб-графом и, как и само веб-пространство, имеют довольно сложную структуру. Например, на сайте Оксфордского университета Google проиндексировал около 8 миллионов страниц, а на сайте Гарвардского университета около 6 миллионов страниц.

В отличие от веб-графа всего веба, веб-граф сайта имеет определенную точку входа – главную страницу сайта, ее можно идентифицировать по доменному имени. Эту страницу так же называют индексной страницей из-за того, что поисковые системы начинают индексирование веб-сайта именно с нее.

Это значит, что у веб-графа сайта будет явная начальная вершина. Соответственно в таком графе можно выделить уровневую структуру. У главной страницы сайта уровень 0, у каждой страницы, ссылка на которую найдена на главной странице, уровень 1 и так далее. То есть уровень страницы определяется минимальным количеством гиперссылок от главной страницы до искомой.

Кроме того, веб-граф сайта будет являться как минимум слабо-связным[4]. В контексте ориентированного графа это означает, что при замене ориентированных ребер (дуг) на неориентированные, будет существовать как минимум один путь из любой вершины в любую другу вершину графа.

Веб-графы некоторых сайтов (обычно это сайты с небольшим количеством страниц) могут быть не только слабо-связными, а еще и сильно-связными. То есть из любой вершины такого графа существует хотя бы один ориентированный путь в любую другую вершину.

Постановка задачи

Требуется проверить предположение о том, что веб-сайты, одинаковые по тематике имеют одинаковую структуру характеристических векторов клик веб-графов. Предлагается проверить это путем сравнения этих векторов.

Для построения характеристических векторов необходимо уметь:

1. Находить все клики заданного размера в веб-графе;
2. Строить веб-граф, на основе списка страниц сайта и гиперссылок между ними;
3. Собирать и записывать в файл все страницы сайта и гиперссылки между ними;

Сравнение характеристических векторов клик веб-графа предлагается производить при помощи кластерного анализа.

Глава 1. Характеристический вектор клик веб-графа

Как было сказано во введении любой веб-сайт может быть описан веб-графом, где вершины графа – страницы сайта, а ребра – гиперссылки между страницами. Такой граф получается ориентированным, так как ссылки имеют явное направление с одной страницы на другую.

Для изучения веб-графов введем понятие характеристического вектора клик веб-графа.

В теории графов клика - это подмножество вершин графа, таких что любые две вершины этого подмножества соединены ребром[9].

Но так как нас интересуют ориентированные графы, то в это определение нужно ввести небольшое дополнение: клика в ориентированном графе – это подмножество вершин графа, таких что любые две вершины этого подмножества соединены дугами (ориентированными ребрами) в обе стороны.

Вообще термин “клика” был введен еще в середине 20 века при изучении социальных сетей, этим термином назывались группы людей, знакомых друг с другом.

Размерность клики – количество вершин, содержащихся в клике.

В математике существует теорема о присутствии хотя бы одной клики в графе. Она формулируется следующим образом: “Граф, состоящий из n вершин и имеющий более чем $[\frac{n}{2}]\bullet [\frac{n}{2}]$ ребер, содержит как минимум одну клику размерности 3”[9].

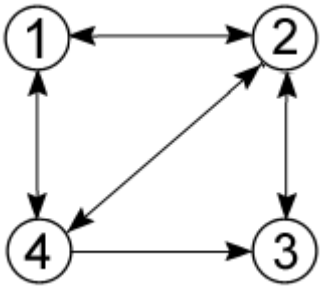

Рис.2.Пример ориентированного графа

На рисунке 2 изображен ориентированный граф. В этом графе вершины 1,2,4 образуют клику размерности 3. Вершины 2,4,3 клику не образуют, так как вершины 3 и 4 связаны только в одну сторону.

Для исследования интересны клики, размерность которых 3 и больше, так как клика размерности 1 это только одна страница, а клик размерности 2 слишком много на любом веб-сайте.

Таким образом получается, что характеристический вектор клик веб-графа – это вектор, показывающий сколько клик размерности 3 и выше найдено в графе.

Например, вектор (57, 45, 20, 2) показывает, что в графе есть 57 клик размерности 3, 45 клик размерности 4, 20 клик размерности 5 и 2 клики размерности 6.

Глава 2. Программа сканирования сайта для построения его веб-графа

Для построения характеристического вектора клик веб-графа сперва необходимо собрать все страницы сайта и гиперссылки между этими страницами.

Для этого была разработана специальная программа-краулер[6][7][8]. Похожие программы, только более сложные, используются различными поисковыми системами для индексации веб-страниц.

Входными данными для программы является ссылка на главную страницу веб-сайта и желаемое количество сканируемых уровней. Возможность ограничения глубины сканирования сайта сделана для того, чтобы на очень больших сайтах время работы программы не стремилось к бесконечности.

На выходе у программы создаются 2 файла: в одном файле содержатся адреса всех найденных страниц, с указанием уровня каждой страницы, во втором файле содержится список гиперссылок, то есть ссылки между страницами в виде: начальная страница, конечная страница, вес. Вес показывает сколько раз одна страница ссылается на другую.

Алгоритм программы состоит в следующем:

1. Программа загружает начальную страницу, введенную пользователем;
2. С помощью регулярных выражений обнаруживаются ссылки;
3. Происходит их нормализация (то есть приведение к общему виду)[10]. Нормализация может проходить по-разному, в зависимости от поставленных задач. В моей программе в процессе нормализации происходит следующее:
    a. Все символы переводятся в нижний регистр;
    b. Добавление конечного слеша: example.com/alice → example.com/alice/;

c. Удаление протоколов прикладного уровня (так как они автоматически добавляются при запросе к странице): http://example.com/ → example.com/;

d. Удаление верхнего доменного имени: www.example.com/ → example.com/;

e. Удаление неиспользуемых параметров в запросе: example.com/display?id=123&fakefoo=fakebar → example.com/display?id=123;

f. Удаление дублированных слешей: example.com/foo//bar.html → example.com/foo/bar.html;

4. Убираются лишние ссылки (например, ссылки вида “mailto:xxx@yyy.ru”, предназначенные для почтовых программ);

5. Нормированные ссылки сохраняются в специальный массив;

6. Программа начинает переходить по всем ссылкам из массива, и повторять действия, приведенные выше.

Программа останавливается либо по достижении уровня, заданного пользователем, либо, когда она перейдет по всем ссылкам из массива.

Глава 3. Программа поиска клик

После завершения сканирования сайта запускается программа поиска клик. В качестве входных данный она принимает список ребер, составленный предыдущей программой и в качестве выходных данных она выводит характеристический вектор клик графа.

Эта программа на основе файла со списком ребер строит матрицу смежности графа. Матрица смежности это квадратная матрица размера n, где n – количество вершин графа, в которой элемент a_{ij} равен единице если существует ребро, соединяющее вершину i с вершиной j, и равен 0 если такого ребра нет[3]. Такая матрица является наиболее удобным, в контексте данной задачи, способом представления графа. Ниже представлен фрагмент кода, выполняющий описанные действия.

```
Пока (не конец файла)
{
        Читать число x; //чтение начальной вершины
        Читать символ tmp_c; //чтение запятой между вершинами
        Читать число y; //чтение конечной вершины
        ms[x][y] = 1; //ms-матрица смежности
        Читать строку tmp_s; //чтение до конца строки
        Если (y > x_max)
        {
                x_max = y; //поиск максимальной вершины
        }
}
```

Фрагмент кода на псевдоязыке

Затем, для сокращения объема данных удаляются ребра-петли, то есть ссылки страницы на саму себя, и ребра, направленные только в одну сторону. Удаление этих ребер не повлияет на конечный результат, так как для поиска клик нужны только двунаправленные ребра, но увеличится скорость поиска клик. То есть граф на рис.2 после обработки будет выглядеть следующим образом:(см. рис.3)

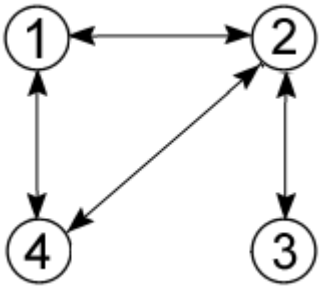

Рис.3. Граф после удаления ребра

Ребро, соединяющее вершину 4 с вершиной 3 было удалено, но количество клик осталось прежним, так как оно было односторонним.

```
Для (i = 0; i <= x_max; i++)
{
	Для (j = 0; j <= x_max; j++)
	{
		Если (ms[i][j] == 0)
		{
			ms[j][i] = 0; //удаление однонаправленных ребер
		}
	}
}
Для (i = 0; i < ms.size(); i++)
{
	Для (j = 0; j < ms[i].size(); j++)
	{
	Если (i == j) пропустить итерацию; //удаление ребер-петель
		Если (ms[i][j] == 1)
		{
			Вывести << i << "," << j << перенос строки;//построение нового
			файла со списком ребер
		}
	}
}
```

Фрагмент кода на псевдоязыке

После подготовки данных они еще раз считываются и запускается программа поиска клик.

Первый шаг – построение числовой последовательности от 0 до dim-1, где dim – размер искомой клики. То есть для клики размера 4 будет построена последовательность 0, 1, 2, 3. Числа этой последовательности являются номерами вершин графа.

```
Для (i = 0; i < dim; i++) //dim-размерность клики
{
    cl_all[0][i] = i; //построение
    последовательности
}
```

Фрагмент кода на псевдоязыке

На следующем шаге проверяется попарная связь вершин по матрице смежности. Если все связи существуют, то эта последовательность запоминается и строится следующая последовательность путем увеличения последнего числа последовательности на единицу.

```
Для (j = 0; j < dim; j++)
{
    Для (k = 0; k < dim; k++)
    {
        Если (j == k) пропустить итерацию; //пропуск диагональных элементов
        матрицы смежности. В клике вершины не должны иметь ребра-петли
        Если (ms[cl_all[i][j]][cl_all[i][k]] == 0)
        {
            tmp = 0; //если между вершинами нет связи, то в переменную
            tmp записывается 0.
        }
    }
}
Если (tmp == 1) //если все связи есть, то есть tmp не равно 0, то
строится следующая последовательность
{
    cl_all[cl_all.size()] = cl_all[cl_all.size() - 1];
    cl_all[cl_all.size() - 1][dim - 1]++;
}
```

Фрагмент кода на псевдоязыке

Если же связь между какой-либо парой отсутствует, то второе число пары увеличивается на единицу и снова происходит проверка связей.

```
Для (j = 0; j < dim - 1; j++)
{
    Для (k = j + 1; k < dim; k++)
    {
        Если (ms[cl_all[cl_all.size() - 1][j]][cl_all[cl_all.size() - 1][k]] == 0)
        {
            cl_all[cl_all.size() - 1][k]++; //увеличение второго числа пары, если между соответствующими вершинами нет связи
        }
    }
}
```

Фрагмент кода на псевдоязыке

После каждого увеличения числа проверяются два условия:

1. cl[i]=x_max – dim + i;
2. cl[0]=x_max – dim + 1 и cl[dim-1] = x_max,

где cl – массив с последовательностью вершин;

x_max – количество вершин графа;

dim – размерность текущей клики;

Первое условие означает что текущее число стало на единицу больше максимально возможного на данной позиции, и в случае выполнения данного условия необходимо перестроить последовательность следующим образом: увеличить на единицу число, стоящее на позиции i-1, далее в цикле каждому следующему элементу присваивать предыдущий, увеличенный на единицу, чтобы все числа были расположены строго по возрастанию.

```
Если (cl_all[cl_all.size() - 1][i] == x_max - dim + i)
{
    cl_all[cl_all.size()] = cl_all[cl_all.size() - 1];
    cl_all[cl_all.size() - 1][i - 1]++; //копирование последовательности для перестроения
```

```
        Для (j = i; j < dim; j++)
        {
            cl_all[cl_all.size() - 1][j] = cl_all[cl_all.size() - 1][j - 1] + 1; //перестроение
            последовательности
        }
    }
```

Фрагмент кода на псевдоязыке

Второе условие означает, что все возможные числовые последовательности при данных размерности клики и количестве вершин построены. При срабатывании этого условия начинается поиск клик следующей размерности. Для этого очищается массив старых клик и на единицу увеличивается переменная, отвечающая за размерность клик.

Программа завершает свою работу если при увеличении размерности она не находит ни одной клики.

Глава 4. Эксперименты с сайтами СПбГУ

Как было сказано выше программы реализуют построение характеристических векторов клик веб-графов. Для проверки предположения о схожей структуре векторов рассмотрим сайты научных подразделений Санкт-Петербургского государственного университета. Далее будут приведены результаты работы обеих программ.

4.1. Сканирование сайтов

Для удобства результаты работы программы сканирования сайтов представлены в таблице (см.таблица1). Данная таблица отображает сколько страниц было найдено на каждом сайте и сколько времени (в секундах) было затрачено на поиск.

Учебно-научное подразделение	**Сайт**	**Количество страниц**	**Время обработки(сек)**
Факультет ПМ-ПУ	apmath.spbu.ru	5713	3679
Факультет свободных искусств и наук	artesliberales.spbu.ru	2062	1897
Факультет искусств	arts.spbu.ru	881	890
Биологический факультет	bio.spbu.ru	52	30
Институт химии	chem.spbu.ru	1842	2984
Факультет стоматологии и медицинских технологий	dent.spbu.ru	68	92
Институт наук о земле	earth.spbu.ru	1902	1896
Экономический факультет	econ.spbu.ru	6123	13555
Факультет военного обучения	fvo.spbu.ru	716	8592
Высшая школа математики	gsom.spbu.ru	3888	12978

Институт истории	history.spbu.ru	1111	1195
Юридический факультет	law.spbu.ru	4417	2542
Математико-механический факультет	math.spbu.ru	1498	819
Восточный факультет	orient.spbu.ru	581	399
Филологический факультет	phil.spbu.ru	4970	1196
Институт философии	philosophy.spbu.ru	4033	3467
Физический факультет	phys.spbu.ru	4657	7130
Факультет политологии	politology.spbu.ru	1621	855
Факультет психологии	psy.spbu.ru	836	1639
Факультет международных отношений	sir.spbu.ru	1130	836
Факультет социологии	soc.spbu.ru	938	1697

Таблица 1. Результаты работы программы сканирования сайтов

Ниже представлен график зависимости времени обработки от количества страниц сайта.

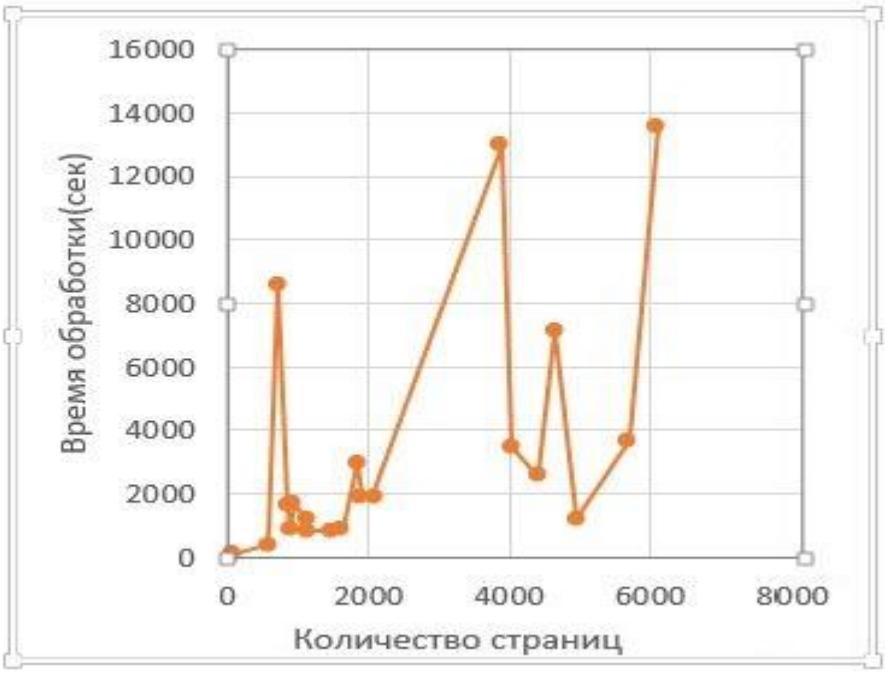

График зависимости времени обработки от количества страниц сайта

Как видно из графика нет прямой зависимости между количеством найденных программой страниц и временем ее работы. Это происходит из-за того, что не только все страницы разные (на некоторых страницах может быть множество изображений, что достаточно сильно влияет на скорость загрузки такой страницы), но еще и они находятся на разных серверах.

4.2. Построение векторов клик веб-графа

Результаты работы этой программы так же представлены в таблице (см.таблицы 2-5). В таблице показаны построенные вектора клик веб-графов сайтов. Крайний левый столбец показывает размерность клик.

	bio.spbu.ru	dent.spbu.ru	fvo.spbu.ru	sir.spbu.ru	orient.spbu.ru	law.spbu.ru
3	0	0	0	59	1309	462
4	0	0	0	43	3189	1366
5	0	0	0	22	5866	3003
6	0	0	0	7	8358	5005
7	0	0	0	1	9406	6435
8	0	0	0	0	8428	6435
9	0	0	0	0	6005	5005
10	0	0	0	0	3367	3003
11	0	0	0	0	1456	1365
12	0	0	0	0	469	455
13	0	0	0	0	106	105
14	0	0	0	0	15	15
15	0	0	0	0	1	1

Таблица 2. Результаты работы программы построения векторов клик веб-графа

	earth.spbu.ru	philosophy.spbu.ru	psy.spbu.ru	soc.spbu.ru	history.spbu.ru
3	16995	17813	2560	50717	16755
4	162832	82318	7951	820390	179390
5	1277185	327745	19842	10470510	1535941
6	8325484	1114095	39624	109731600	10740575
7	45863995	3234075	63744	970788807	62894930
8	216637901	8042651	83107	6276343795	314460497
9	786828856	17219277	88047	26580943606	1302872731
10	1772882779	31916974	75715	65444941253	4213490413
11	2913200983	51489647	52584	126607820000	10215607505
12	3881840310	70247325	29212	170452108066	16584017224
13	4791433131	83892868	12782	191677315918	18988699722
14	5008676709	100029242	4301	194062797618	16056844485
15	4224918978	109355969	1072	159407063220	10974050363
16	2882831215	97681125	357	97611321092	6549905855
17	1508009009	60899298	102	65760747020	3167337974
18	638656895	35004916	19	28044328174	1262047987
19	220100326	14943599	1	11007398808	332511783
20	64761219	4485411	0	2765498877	79651868
21	14516227	862746	0	537480238	13027859
22	2356971	110387	0	55064850	2033649
23	295175	12290	0	5085239	169728
24	1277	298	0	6306	9074
25	233	125	0	26	204
26	27	27	0	1	27
27	1	1	0	0	1

Таблица 3. Результаты работы программы построения векторов клик веб-графа

	gsom.spbu.ru	econ.spbu.ru	chem.spbu.ru	arts.spbu.ru	artesliberales.spbu.ru
3	907270	521931	16361	6174	74560
4	32216937	18698850	146758	42635	1111296
5	927100280	534257751	1116642	241215	13307693
6	19953979326	11498829575	7127377	1114830	132446812
7	296247552229	170717833235	38446045	4282768	1123660175
8	2309457042913	1330865012550	177426210	13896548	7264687763
9	16084213575367	9268809379908	709184516	38578321	30766679147
10	93862162635361	54089712834047	1597934551	92568981	75750640727
11	397562576058336	229102387679888	2625726055	152109350	146544917032
12	1440528238089770	830129591519308	3498779968	202685708	197293421800
13	2919518580136550	1682423643132180	4318614090	250179023	221860990563
14	4415333924669510	2544413396690950	4514420053	261522140	224622117149
15	5569722979274350	3209650279255800	3808003603	220599156	184509099468
16	5710787353170430	3290941091878510	2598353179	150523628	112982301968
17	4937718069171750	2845446396270920	1359198548	78738910	76116176836
18	3457072202990400	1992198319920980	575634177	33346716	32460504774
19	2044361876212640	1178099286320470	198380806	11492279	12740748124
20	1010665047657610	582413409880391	64020581	3381431	3200985559
21	307504947400306	177205104090208	14860713	757948	622117945
22	77521997239617	44673406741141	2332151	123066	63735983
23	11343018636101	6536612874364	290446	15412	5886018
24	1047595829156	603695418625	14299	372	7299
25	32737369661	18865481832	219	191	26
26	65474739	37730964	27	27	1
27	99620	57408	1	1	0
28	1463	914	0	0	0
29	30	30	0	0	0
30	1	1	0	0	0

Таблица 4. Результаты работы программы построения векторов клик веб-графа

	apmath.spbu.ru	politogy.spbu.ru	phys.spbu.ru	phil.spbu.ru	math.spbu.ru
3	146902	6728	8105459	9066	15483
4	1912263	52696	735482352	37015	160625
5	22030738	325095	59025893451	123426	1297954
6	220832538	1623622	1797802281041	338515	8447815
7	1882266138	6724850	38694098494846	774957	45551773
8	12169227033	23535985	574473481067667	1499963	207748229
9	51537893408	70607515	2755002473156210	2478339	1446862541
10	126891447359	159033542	8167204831671580	3518492	8443412324
11	245480387318	261323917	47661112220040900	4308217	35762917239
12	330490245446	348214119	106551182479123000	4555330	129583354324
13	371643882280	429807652	386077554594856000	4154952	262626584208
14	376269101888	449295131	512209091680995000	3258767	397183314627
15	309074965673	378989429	774639419803753000	2001649	501026892237
16	189258964480	258599647	822241012150694000	1003539	513716400336
17	127503764370	135273475	843065910265435000	496117	444174611223
18	54375255353	57289670	728940077992803000	200560	310982457933
19	21342287726	19743739	510356898869538000	62595	183901476116
20	5362036368	5809299	250767143854407000	15793	90914821043
21	1042122492	1302154	123971000605872000	3399	27661743451
22	106765449	211428	25322316583755400	518	6973525524
23	9859789	26478	6383756010764720	281	1020366255
24	12226	372	604107598810687	162	94236946
25	26	191	43710809419546	26	2944905
26	1	27	1365962794361	1	58898
27	0	1	2731925589	0	589
28	0	0	4156625	0	316
29	0	0	14634	0	30
30	0	0	254	0	1
31	0	0	32	0	0
32	0	0	1	0	0

Таблица 5. Результаты работы программы построения векторов клик веб-графа

Почти у всех векторов (кроме первых четырех) имеется схожая структура: количество клик с увеличением размерности сперва увеличивается, затем начинает уменьшаться и приходит к единице. Так же можно заметить, что возрастание сменяется на уменьшение в среднем на клике размера $\frac{\max\_dim}{2}$ (max_dim – максимальная размерность клики в данном векторе).

Глава 5. Кластерный анализ

5.1. Общие сведения

Кластерный анализ используется в различных областях науки. Например, его широко используют в статистике, социологии, маркетинге, биологии и так далее[11].

Кластерный анализ применяется для того, чтобы разбить некоторые данные на группы так, чтобы в одной группе находились схожие данные. В качестве примера можно рассмотреть точки случайным образом расположенные на плоскости(см.рис.4).

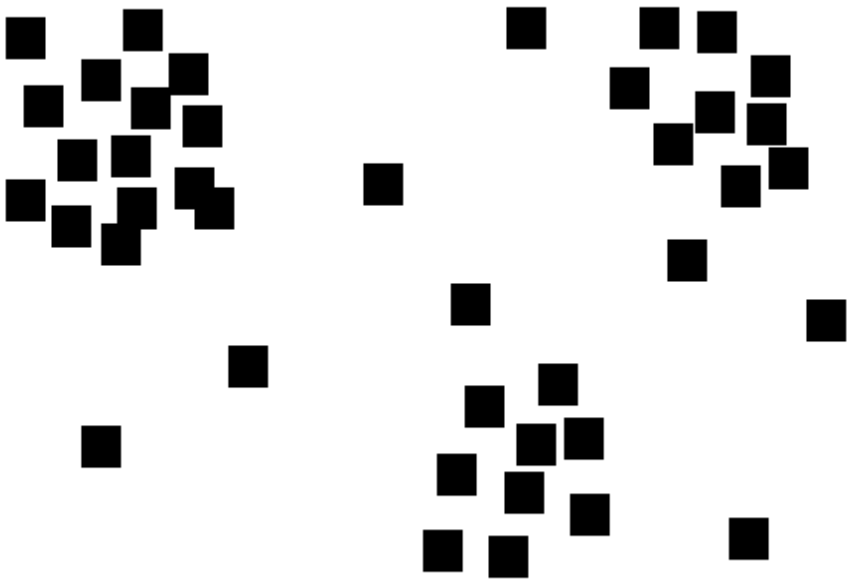

Рис.4.Точки на плоскости

В процессе кластеризации эти точки могут быть разбиты на три кластера (в зависимости от используемого алгоритма) следующим образом:

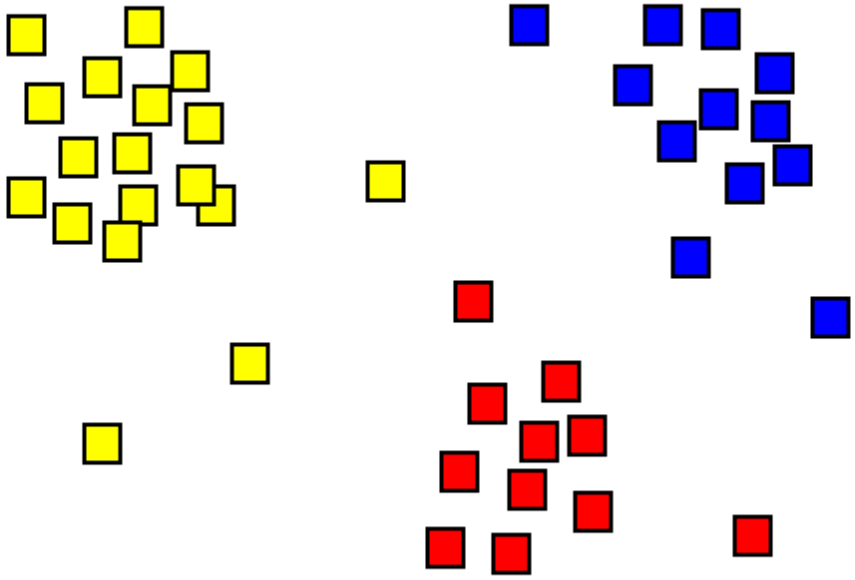

Рис.5.Точки на плоскости, разбитые на кластеры

На сегодняшний день существует большое количество различных методов кластеризации. Подробно рассмотрим метод под названием К-средних (K-means) так как именно он был выбрал для обработки результатов работы программы поиска клик.

Идея данного метода состоит в следующем: множество элементов векторного пространства разбивается на заранее выбранное число кластеров k. Случайным образом выбирается k элементов исходного множества, эти элементы будут служить начальными центрами кластеров. После этого происходит вычисление расстояний от каждого элемента до каждого центра кластера и, исходя из полученных результатов, элементы попадают в определенные кластеры. Далее вычисляется центр масс кластера, как среднее значение среди всех элементов кластера, снова вычисляются расстояния между элементами и центрами масс и элементы еще раз распределяются по

кластерам. Последняя итерация происходит до тех пор, пока центры масс не перестают изменяться.

Чаще всего под расстоянием в данном методе понимается евклидово расстояние, то есть корень из квадрата разности соответствующих координат, при этом кластеры стремятся принять форму эллипса. Так же можно использовать расстояние Манхэттена – это расстояние между точками, пройденное по линиям, параллельным осям координат, в этом случае кластеры принимают прямоугольную форму. Эти две метрики являются наиболее распространенными для алгоритма кластеризации к-средних, но не исключено использование других метрик[12].

5.2. Кластеризация результатов работы программы построения векторов клик веб-графа

Кластеризация производилась в программном пакете для статистического анализа данных Statistica. В ней есть возможность проводить кластерный анализ векторов методом к-средних с евклидовой метрикой.

При использовании метода к-средних необходимо заранее выбрать количество кластеров, на которое будет разбито исходное множество данных. Обычно это число выбирается либо исходя из каких-то наблюдений за данными, либо перебором нескольких вариантов и анализом полученных результатов. Во втором случае необходимо следить за двумя факторами:

1. Не должно быть много кластеров, содержащих всего один элемент множества;
2. Расстояние от центра масс кластера до элементов множества, содержащихся в этом кластере должно быть примерно одинаковым и не большим.

Так как полученные характеристические вектора клик веб-графа представляют из себя довольно сложную структуру, то количество кластеров будем выбирать вторым способом. Самым оптимальным разбиением получилось разбиение исходного множества на 7 кластеров следующим образом (см.рис. 6):

Case name	Cluster members (analyse.sta) Number of clusters: 7 Total number of training cases: 21 Case No.	Final classification
bio.spbu.ru	1	1
dent.spbu.ru	2	1
fvo.spbu.ru	3	1
sir.spbu.ru	4	1
law.spbu.ru	6	1
orient.spbu.ru	5	2
psy.spbu.ru	9	2
earth.spbu.ru	7	3
philosophy.spbu.ru	8	3
soc.spbu.ru	10	3
history.spbu.ru	11	3
chem.spbu.ru	14	3
math.spbu.ru	21	3
arts.spbu.ru	15	4
politology.spbu.ru	18	4
phil.spbu.ru	20	4
phys.spbu.ru	19	5
gsom.spbu.ru	12	6
econ.spbu.ru	13	6
artesliberales.spbu.ru	16	7
apmath.spbu.ru	17	7

Рисунок 6. Скриншот окна программы Statistica с результатами кластеризации

Анализируя результаты кластеризации видно, что не только все вектора не попали в один кластер, но еще и в большинстве случаев вектора сайтов сходных учебно-научных подразделений (например, сайт математико-механического факультета и сайт факультета прикладной математики-процессов управления) оказались в разных кластерах.

Это происходит из-за того, что сайты всех подразделений делают разные люди, исходя из своих предпочтений. Однако, если бы все сайты какой-либо организации (в данном случае Санкт-Петербургского государственного университета) создавала бы одна команда разработчиков, или существовал некий шаблон, то пользователи могли бы быстрее находить интересующую их информацию[5]. Кроме того, если бы имелась схожая структура, то поисковые системы более эффективно индексировали страницы сайтов, что в свою очередь привело бы к увеличению рейтинга сайта, и как следствие более высокие места в поисковой выдаче.

Выводы

В рамках выпускной квалификационной работы были поставлены и решены следующие задачи:

1. Разработан и реализован, на языке C++, собственный алгоритм, позволяющий получать файл со списком страниц сайта и файл со списком гиперссылок между этими страницами. Реализация алгоритма показала, что скорость выполнения программы очень сильно зависит от размера страницы, настроек и местоположения сервера, на котором размещен сайт.
2. Получены файлы со списками всех страниц и гиперссылок сайтов научных подразделений Санкт-Петербургского государственного университета.
3. Разработан и реализован, на языке C++, собственный алгоритм, позволяющий строить характеристические вектора клик веб-графов. Во время работы данной программы была выявлена сильная нагрузка на оперативную память при увеличении количества страниц сайта.
4. Получены файлы, содержащие характеристические вектора клик веб-графов научных подразделений Санкт-Петербургского государственного университета.
5. Описан процесс кластеризации методом к-средних.
6. Проведена кластеризация характеристических векторов клик веб-графов научных подразделений Санкт-Петербургского государственного университета.
7. Опровергнута гипотеза о схожести (т.е. попадании в один кластер) характеристических векторов клик веб-графов сайтов схожих организаций (в данном случае сайтов научных подразделений Санкт-Петербургского государственного университета).

Заключение

Выпускная квалификационная работа была посвящена вопросам сбора и анализа страниц и гиперссылок сайтов научных подразделений Санкт-Петербургского государственного университета, с целью подтверждения или опровержения гипотезы о схожести характеристических векторов у схожих организаций.

Все задачи, поставленные в ходе исследования, были решены.

Список литературы

1. Райгородский А.М. Модели случайных графов. // Труды МФТИ. 2010. Том 2, № 4.
2. Печников А.А., Чернобровкин Д.И. Об исследованиях веб-графа сайта // Материалы конференции «Управление в технических, эргатических, организационных и сетевых системах». – СПб.: «Концерн «ЦНИИ «Электроприбор», 2012, С. 1069-1072.
3. Харари Ф. Теория графов. — М.: Мир, 1973.
4. Broder Andrei, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, Janet Wiener (2000). "Graph structure in the web". Proceedings of the 9th World Wide Web Conference.
5. Чернобровкин Д.И. Об исследованиях веб-графа сайта с использованием имитационной модели.
6. Печников А.А., Чернобровкин Д.И. Адаптивный краулер для поиска и сбора внешних гиперссылок. // Журнал “Управление большими системами: сборник трудов”. Выпуск №36/2012.
7. Filippo Menczer, Gautam Pant, Padmini Srinivasan. Topical web crawlers: evaluating adaptive algorithms.
8. Mike Thelwall. Big data and social web research methods.
9. Клика. https://ru.wikipedia.org/wiki/Клика_(теория_графов) (Дата обращения 05.05.2016).
10. Network Working Group. RFC 3986 — Uniform Resource Identifier (URI): Generic Syntax. — WWW, 2005.
11. Кластерный анализ. https://ru.wikipedia.org/wiki/Кластерный_анализ (Дата обращения 05.05.2016).
12. www.basegroup.ru. Модуль М.123 Алгоритм кластеризации k-means.