МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
МОСКОВСКИЙ ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ (государственный университет)
ФАКУЛЬТЕТ УПРАВЛЕНИЯ И ПРИКЛАДНОЙ МАТЕМАТИКИ
ВЫЧИСЛИТЕЛЬНЫЙ ЦЕНТР ИМ. А. А. ДОРОДНИЦЫНА РАН

Лексин Василий Алексеевич

Методы выявления взаимосогласованных структур сходства в системах взаимодействующих объектов

511656 — Математические и информационные технологии

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

Научные руководители:

чл.-корр. РАН, д.ф.-м.н. К. В. Рудаков
к.ф.-м.н. К. В. Воронцов

Москва 2005 г.

Содержание

Аннотация

В работе развиваются и исследуются алгоритмы обработки данных, предназначенные для решения задач Анализа Клиентских Сред (АКС). Клиентская среда — это совокупность двух множеств: множества клиентов и множества ресурсов, которыми эти клиенты регулярно пользуются. Предполагается, что все факты пользования протоколируются в электронном виде. Технология АКС предназначена для выявления структур сходства между клиентами и между ресурсами на основе имеющихся данных. Рассматривается частная прикладная задача — применение технологии АКС для обработки логов поисковой машины Яндекс. В основе анализа лежит принцип «схожи те пользователи, которые посещают схожие множества ресурсов, и схожи те ресурсы, на которые заходят схожие пользователи». Предлагаемые методы оценивания сходства позволяют строить достаточно адекватные Карты Интернета, на которых близким по тематике сайтам, оказывается, соответствуют близкие точки. Рассматриваются и другие перспективные возможности, открывающиеся с применением технологии АКС: персонализация поиска, направленное предложение ресурсов пользователям, поиск схожих ресурсов, навигация в сети Интернет по картам сходства.

1 Введение

Клиентская среда – это совокупность клиентов компании, регулярно пользующихся услугами данной компании, действия которых компания тщательно протоколирует. Технология анализа клиентских сред (АКС) – это цепочка процедур обработки данных, ведущая от исходных протоколов действий клиентов к решению широкого спектра задач, возникающих у современных компаний при управлении взаимоотношениями с клиентами (Customer Relationship Management, CRM). К числу этих задач относятся: выявление и интерпретация типов поведения клиентов (сегментация клиентской базы), выявление целевых групп потенциальных и существующих клиентов, структуризация ассортимента в соответствии с объективными предпочтениями клиентов, прогнозирование возможного оттока клиентов, выявление необычного или потенциально опасного для компании поведения клиентов. Конечной целью этой деятельности является более эффективное привлечение и удержание клиентов, в первую очередь за счет повышения качества оказываемых услуг.

Технология АКС основана на понятии сходства. Клиенты схожи с точки зрения компании, если они пользуются схожими услугами. И, наоборот, услуги схожи, если ими пользуются схожие клиенты. Данное определение является рекурсивным и позволяет организовать итерационный вычислительный процесс, в котором обе меры сходства уточняются поочередно. Оказалось, что такого рода процессы достаточно быстро сходятся и приводят к парам взаимосогласованных мер сходства (метрик). Метрика в пространстве клиентов позволяет решать задачи сегментации, поиска схожих клиентов, обнаружения аномалий в поведении клиентов. Метрика в пространстве услуг позволяет объективно позиционировать услуги и структурировать ассортимент услуг.

Технология АКС является общей для огромного числа компаний, работающих в самых разных сферах бизнеса. Можно говорить о клиентских средах операторов связи, торговых сетей, эмитентов пластиковых (в частности, кредитных) карт, электронных магазинов, библиотек, организаторов биржевых торгов и т.п.

Идея данного исследования заключается в применении технологии АКС к клиентской среде поисковой машины. Здесь в роли «услуг» выступают ресурсы URL, предлагаемые в качестве результатов поиска. Клиентами являются пользователи поисковой машины. Пользование услугой – это переход пользователя со страницы результатов поиска на соответствующий ресурс.

Хотя технология апробирована на исходных данных поисковой машины Яндекс, она универсальна и может быть использована для решения широкого класса задач Анализа Клиентских Сред.

Целью работы является разработка, реализация и экспериментальная проверка несколько упрощенного варианта АКС, в котором мера сходства строится только на множестве услуг (в данном случае — ресурсов сети Интернет).

Актуальность работы очевидна в связи с растущим спросом на системы управления взаимодействия с клиентами (CRM).

Новым в данной работе, по сравнению с ранее реализованными методами АКС, является применение техники проверки статистических гипотез для оценивания расстояний между услугами (ресурсами), которая приводит к существенной разреженности матрицы попарных расстояний, и как следствие, к повышению точности карт сходства и эффективности их построения.

1.1 Исходные данные

Исходными данными для анализа являются протоколы переходов пользователей на ресурсы, найденные поисковой машиной Яндекс.

Данные, предоставленные Яндексом, охватывают 7 дней работы поисковой машины, по 5–10 миллионов запросов в день. Внутри файла данных хранятся упорядоченные по времени первого обращения группы записей. Каждая группа записей относится к отдельному пользователю и состоит из заголовка и описаний отдельных запросов данного пользователя. Заголовок включает зашифрованный уникальный идентификатор пользователя и время первого запроса. Описание запроса включает текст запроса, его время, номер страницы, общее число найденных документов. Затем следуют данные о найденных документах: URL документа, время выбора документа пользователем (если документ не был выбран, то пусто).

В данной работе выявление структур сходства пользователей и ресурсов ведется только на основе информации типа «пользователь X посетил ресурс Y», без попытки проанализировать тест запроса и время захода на ресурс. Хотя по разности времени захода на ресурсы в одном запросе можно было бы судить о том, сколько пользователь находился на каждом из них. Таким образом, из всей информации, содержащейся в логе, учитываются только два типа фактов:

1. ресурс был выдан в качестве результата поиска и пользователь зашел на него;
2. ресурс был выдан в качестве результата поиска, но пользователь проигнорировал его.

Исследуемый лог содержал данные о 14606 пользователях и 207312 запросах. Из 1972636 документов, предлагавшихся поисковой машиной в качестве результатов поиска, 129600 были выбраны пользователями. Для наглядности приведем фрагмент лога:

```
1098353321109615996
        французская кухня       1110473322      113906  0
                http://www.naturel.ru/
                http://www.kuking.net/c7.htm    1110473328
                http://www.cooking-book.ru/national/french/
            ...
        жаренное мясо в вине    1110473174      1349    0
            ...
      ...
...
```

1.2 Постановка задачи

Для выявления предпочтений и информационных потребностей огромного числа пользователей по отношению к огромному числу ресурсов простейшая тактика «пользователи ресурса *X* посещают также ресурс *Y*» не подходит. Предлагается применить более тонкий анализ, основанный на принципе АКС «схожи те пользователи, которые посещают схожие множества ресурсов, и наоборот, схожи те ресурсы, на которые заходят схожие множества пользователей».

В рамках данного исследования ставится задача разработать и реализовать специальный вариант АКС, приспособленный для решения задач персонализации услуг и ресурсов Интернет, а также задача выявления и визуализации кластерных структур ресурсов и пользователей.

2 Алгоритмы обработки данных

Общая схема обработки включает в себя следующие стадии:

3. Формирование полных словарей пользователей и ресурсов и подсчет числа посещений n_u, n_r (однократное чтение логов, первый проход).
4. Построение гистограмм числа посещений, определение порогов отсечения, редукция словарей.
5. Формирование матрицы посещений (второй проход по логам).
6. Формирование разреженной матрицы сходства ресурсов.
7. Построение карты сходства всех ресурсов (карта Интернета).
8. Выбор ресурса, выделение значимо близких к нему ресурсов и построение карты сходства ближайшей окрестности данного ресурса.

Далее перечисленные стадии обработки данных рассматриваются подробно.

2.1 Фильтрация исходных данных и создание словарей

В связи с большим объемом исходных данных и наличием большого числа неинформативных пользователей и ресурсов, характеризующихся малым числом посещений, возникает необходимость эффективной фильтрации исходных данных и формирования словарей информативных ресурсов и пользователей.

2.1.1 Фильтрация ресурсов

Основной вопрос, возникающий при анализе исходных данных: что считать ресурсом? Неправильно было бы считать ресурсом, например доменное имя, т. к. внутри домена может содержаться множество сайтов различной тематики, более того, отдельные сайты могут содержать разделы разной направленности. В таких случаях логичнее ресурсом считать не весь сайт, а каждый его раздел. С другой стороны, считать ресурсом каждый документ, выдаваемый в качестве результата поиска тоже неправильно. Их будет слишком много, а число заходов на них будет невелико. Очевидно, не имеет смысла говорить о сходстве ресурсов, на которые сходило слишком мало пользователей.

Для решения этого вопроса делается естественное предположение, что сайт имеет древовидную структуру. Узлы дерева соответствуют каталогам и подкаталогам сайта. В вершине дерева стоит доменное имя сайта. Листьям дерева соответствуют документы, выдаваемые в результатах поиска. Пример дерева ресурсов сайта показан на рис.1.

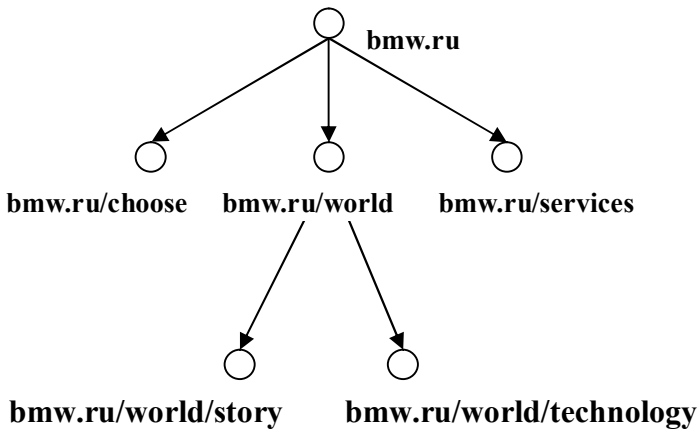


Рис.1. Пример построения дерева ресурсов.

При анализе исходных данных, как уже отмечалось, нас интересуют ресурсы, на которые сходило достаточное количество пользователей. Наиболее простой способ отбора ресурсов состоит в установлении порога на число посещений. Для этого в процессе построения дерева ресурсов подсчитывается число посещений для каждого узла дерева. Например, если пользователь зашел на ресурс «moscowout.ru/sport/billiards», то счетчик посещений увеличивается на единицу для узлов «moscowout.ru», «moscowout.ru/sport» и «moscowout.ru/sport/billiards». После того, как дерево построено, выбирается порог отсева ресурсов по числу посещений и все узлы дерева с меньшим числом посещений отбрасываются. Если число посещений для узла больше заданного порога, то он считается ресурсом. На рисунке 2 показан пример «обрезания» дерева ресурсов.

Если посещаемость вершины дерева (домена) меньше заданного порога, то этот домен целиком исключается из рассмотрения.

Гистограмма посещаемости документов показывает, что около 100 тысяч документов, (что составляет основную массу — примерно 77% документов) были выбраны только один раз, около 10 тысяч — два раза, далее гистограмма очень быстро убывает (рис. 3). Отбрасывание документов с малым числом посещений необходимо, поскольку для них невозможно построение статистически значимых оценок сходства. Очевидно, что увеличение общего числа документов со значимым числом посещений возможно только путем расширения анализируемого интервала времени.

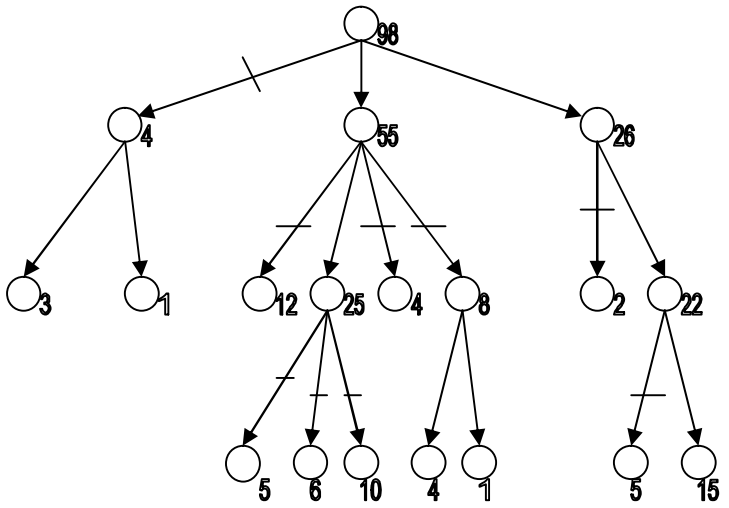


Рис.2. Выбор ресурсов из дерева по порогу посещаемости (в данном случае 13 посещений).

В данном исследовании был выбран порог 20 посещений, после чего для дальнейшего анализа осталось 1843 ресурса.

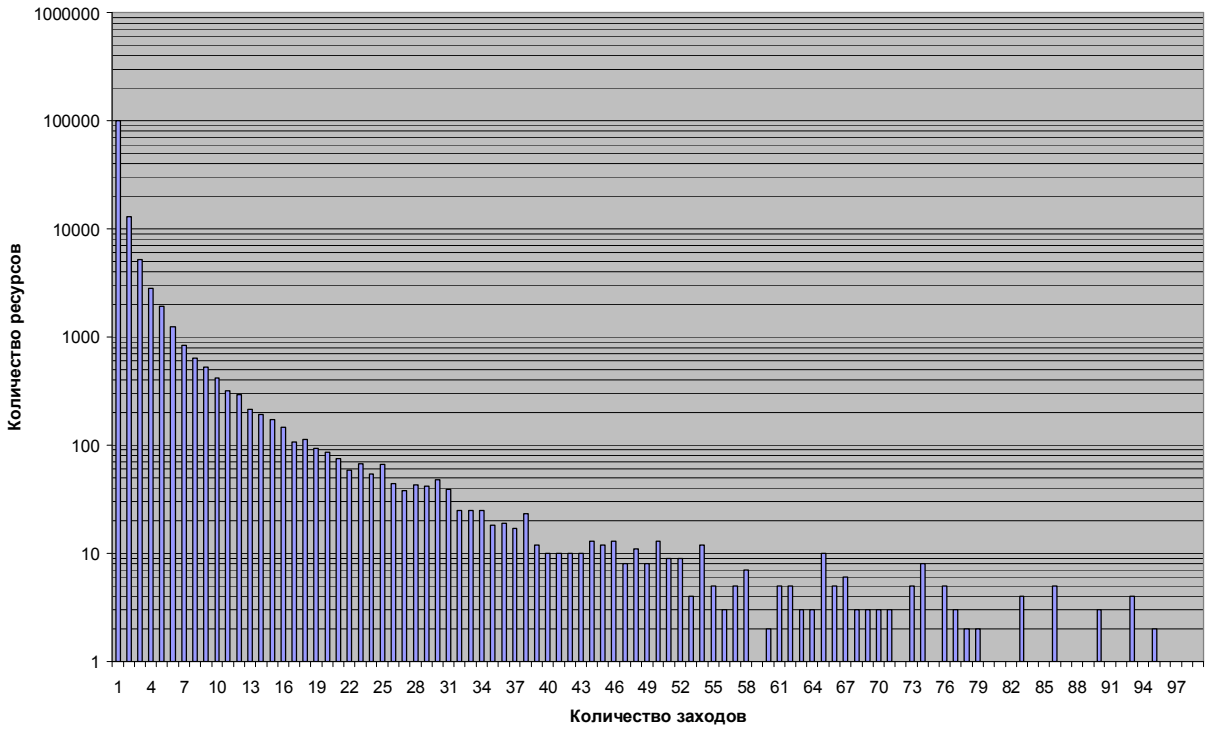


Рис.3. Гистограмма посещаемости ресурсов

2.1.2 Построение словаря ресурсов

Важный вопрос, возникающий при реализации описанной выше идеи, — какую эффективную структуру данных использовать для хранения и быстрого поиска ресурсов, а также хранения счетчиков посещений в узловых точках дерева. Структура данных должна выполнять быстрый логарифмический поиск ресурсов. Примером такой структуры является тернарное дерево поиска (Ternary Seach Tree, TST), которое строит словарь строк побуквенно и реализует быстрый поиск строки (названия ресурса) [1]. В целях данного исследования пришлось несколько модифицировать TST так, чтобы некоторые узлы содержали не только символ, но и счетчик посещений. Эти узлы соответствуют узлам дерева ресурсов (обычно это символ «/»). При добавлении нового ресурса в TST просматриваются все буквы имени ресурса и, как только встречается узловая точка, для нее увеличивается счетчик числа посещений. Построенная структура также дает возможность «обрезать» ненужные поддеревья согласно описанному выше критерию (рис.4).

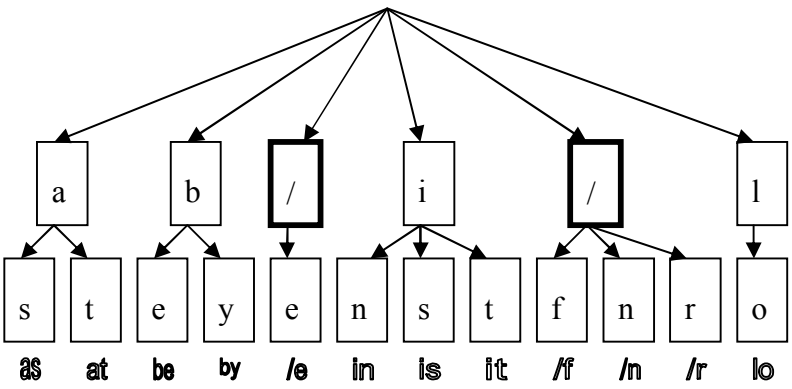


Рис.4. Пример тернарного дерева поиска, содержащего узлы со статистикой посещений (эти узлы выделены жирной линией).

2.1.3 Фильтрация и словарь пользователей

При фильтрации пользователей применяются более простые соображения. Во-первых, нас не интересуют «случайные» пользователи, которые зашли на один или несколько ресурсов. Их поведение неинформативно. С другой стороны, можно предположить, что пользователи, которые ходят на очень многие ресурсы, скорее всего интересуются многими областями и по статистике их посещений сложно судить о сходстве этих ресурсов. В данном исследовании выставлялся только нижний порог на количество посеще-

ний, то есть отбрасывались все пользователи с малым числом посещений. Для выбора порога строилась гистограмма посещений (рис.5).

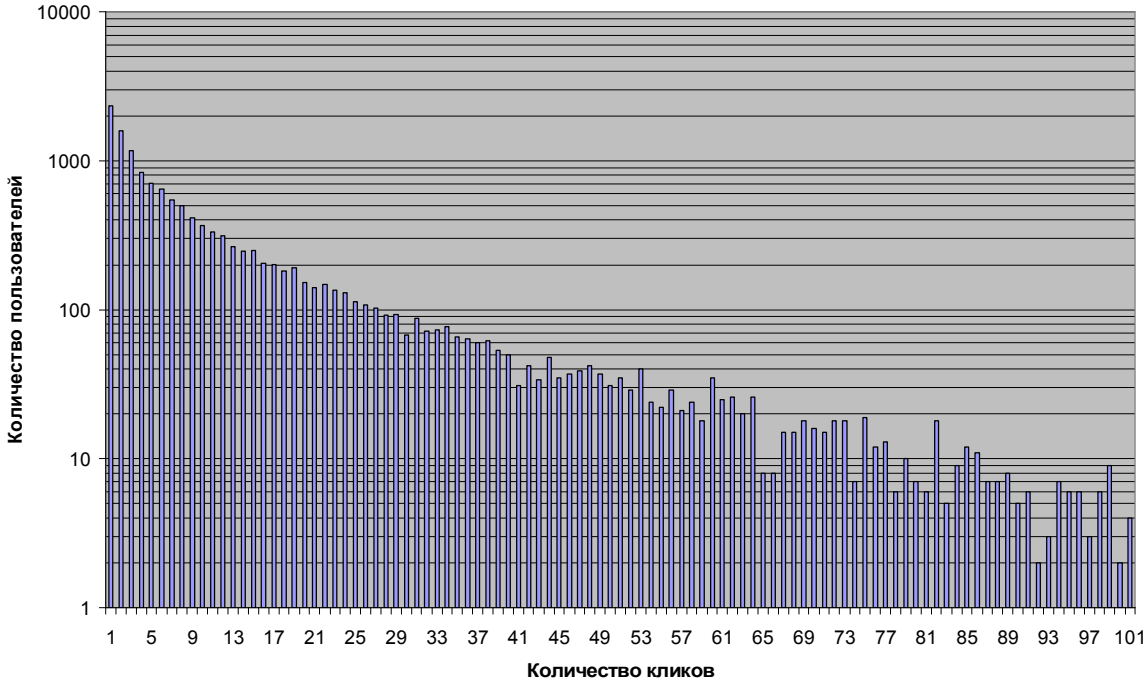


Рис.5. Гистограмма посещений ресурсов пользователями.

Для хранения списка пользователей подходят различные структуры данных. Поскольку в имеющихся данных пользователи представляются целочисленными идентификаторами (набор из 8-10 цифр), для хранения списка использовалась хэш-таблица [2]. Для работы с хэш-таблицей была подобрана достаточно эффективная хэш-функция и задан оптимальный размер таблицы. Хэш-таблица обеспечивает быстрый поиск пользователей и хранение дополнительной информации о каждом пользователе (счетчик посещений).

2.1.4 Построение матрицы посещений

Следующим этапом после формирования словарей пользователей и ресурсов является построение матрицы посещений $A=\|a_{ur}\|_{r=1,R}^{u=1,U}$, где a_{ur} — количество посещений пользователем u ресурса r. Особенность исходных данных заключается в том, что матрица A сильно разрежена и почти не содержит элементов, отличных от нуля и единицы. Для ее хранения используется подходящая структура данных, обеспечивающая эффективный перебор и поиск ненулевых элементов в строках и столбцах. Ячейки такой структуры содержат информацию только о ненулевых элементах и имеют указатели на следующий ненулевой элемент в строке и столбце. Так

щий ненулевой элемент в строке и столбце. Так как почти все ненулевые элементы матрицы равны единицы (пользователи редко ищут один и тот же ресурс дважды), то матрицу можно строить как бинарную, что еще сократит затраты памяти. В исследуемой задаче для хранения матрицы посещений была использована структура данных Sparse Matrix [3]. Заполнение матрицы происходит во время второго прохода по логам поисковой системы. При этом, если пользователь и ресурс содержатся в построенных словарях, то в матрицу A добавляется соответствующий элемент.

2.2 Вычисление оценок сходства ресурсов

Сформулируем задачу вычисления оценок сходства ресурсов. Пусть ресурсы R_i и R_j посещались n_i и n_j пользователями соответственно. Пусть n_{ij} пользователей посетили оба ресурса. Если предположить, что посещения ресурсов R_i и R_j являются независимыми событиями, то количество чисто случайных посещений обоих ресурсов одним и тем же пользователем будет подчиняться гипергеометрическому распределению:

$$P_{ij} = P(n_{ij} = x) = \frac{C_{n_i}^{x} C_{U-n_i}^{n_j - x}}{C_U^{n_j}},$$

где U-количество пользователей.

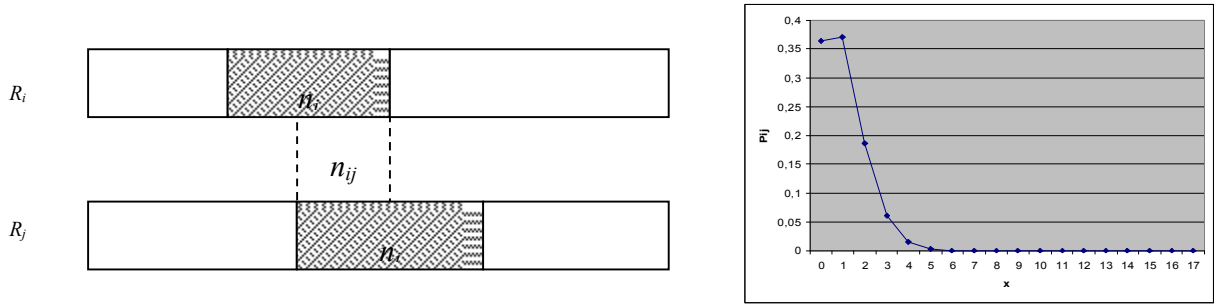


Рис.6. Построение оценки сходства двух ресурсов. График гипергеометрического распределения при $U=10^4$, $n_i=100$, $n_j=100$.

Эта вероятность максимальна при $x \approx \frac{n_i n_j}{U}$ и быстро убывает по мере увеличения x. Если n_{ij} настолько велико, что $P_{ij} < \alpha$ при заданном достаточно малом уровне значимости α, то приходится признать, что либо реализовалось маловероятное событие, либо исходная гипотеза о независимости ресурсов неверна, следовательно имеется статисти-

чески значимая взаимосвязь в посещениях данной пары ресурсов, следовательно они близки.

Если же $P_{ij} > \alpha$, то наблюдаемое распределение посещений (n_i, n_j, n_{ij}) вполне могло реализоваться случайно, и делать какие-либо выводы о сходстве ресурсов R_i и R_j невозможно.

Введем множество пар ресурсов $D = \{(i, j) \mid P_{ij} < \alpha\}$. Чем меньше вероятность P_{ij}, тем более схожи ресурсы. Расстояние между ресурсами будем оценивать по формуле $\rho(i, j) = M(P_{ij})$, где M — некоторая монотонно возрастающая функция. Таким образом, функция расстояния определяется только на подмножестве пар D, и, вообще говоря, с точностью до произвольного монотонного преобразования M.

Монотонная функция M выбирается эвристически таким образом, чтобы построить метрику, наиболее удобную для визуализации и интерпретации карты сходства ресурсов.

Критерием выбора является интуитивная «правильность» получаемой в итоге карты сходства ресурсов. Другим, более формальным, критерием является близость распределения расстояний к равномерному и наличие как больших, так и малых (приближающихся к нулю) расстояний. Только в этом случае возможно образование кластерных структур на карте сходства.

Достаточно интересные карты сходства давала функция $\rho(i, j) = \left(\frac{|\ln \alpha|}{|\ln P_{ij}|} \right)^3$, где α — уровень значимости. Видно, что функция $\rho(i, j)$ лежит в интервале [0,1], является монотонно возрастающей функцией P_{ij} и имеет максимум при $P_{ij} = \alpha$, равный единице.

2.3 Построение карты сходства всех ресурсов

После того, как построена мера сходства ресурсов, применяется алгоритм многомерного шкалирования, позволяющий разместить все ресурсы на плоскости и выявить группы сильно схожих ресурсов.

2.3.1 Алгоритм многомерного шкалирования

Задача многомерного шкалирования хорошо известна в прикладном анализе данных и заключается в следующем. Имеется конечная совокупность объектов, описанная матрицей попарных расстояний (такие совокупности будем называть конечными метрическими конфигурациями). Требуется построить представление данной конфигурации в евклидовом пространстве невысокой размерности так, чтобы евклидовы расстояния между объектами как можно точнее приближали исходные расстояния. Многомерное шкалирование часто используют как средство разведочного анализа данных для наглядного представления метрической конфигурации на плоском точечном графике (карте сходства). Такое представление, несмотря на возможные искажения, отражает наиболее существенные особенности исходной конфигурации, в частности, ее кластерную структуру, если таковая имеется. Обычно задачу многомерного шкалирования решают путём минимизации функционала стресса [4]:

$$S = \sum_{(i,j)\in D} w_{ij}(\rho_{ij} - d_{ij})^2 ,$$

где суммирование ведется по всем парам точек (i, j), для которых известны исходные расстояния ρ_{ij}, $w_{ij} = \rho_{ij}^{\alpha}$ — веса объектов, $d_{ij}^2 = \sum_{k=1}^{n}(x_{ik} - x_{jk})^2$ — евклидовы расстояния между *i*-м и *j*-м объектами, x_{ik} — искомые координаты *i*-й точки, представляющей *i*-й объект в евклидовом пространстве размерности *n*. Показатель степени α позволяет ориентировать процесс размещения точек на более точное отражение далеких (при $\alpha > -2$) или близких (при $\alpha < -2$) расстояний. Принято считать, что наиболее адекватный результат достигается при $\alpha = -2$. В этом случае функционал стресса приобретает смысл потенциальной энергии в системе точек, связанных упругими связями, и задача минимизации приобретает физический смысл поиска устойчивого равновесия.

С решением данной оптимизационной задачи связаны две проблемы.

Во-первых, функционал стресса имеет огромное количество локальных минимумов. Ни один из известных эффективных методов многомерного шкалирования не гарантирует достижения глобального минимума стресса. Используемый в данном исследовании алгоритм не является исключением. В то же время, он ориентирован на поиск такого локального минимума, при котором сохраняются наиболее существенные структурные особенности исходной метрической конфигурации. Для построения карт сходства эта

стратегия представляет даже больший интерес, чем борьба за глобальную минимизацию стресса.

Во-вторых, большинство известных алгоритмов имеют квадратичную по числу объектов сложность, позволяя размещать до нескольких тысяч объектов за приемлемое время. Однако они практически бесполезны для сверхбольших конфигураций, насчитывающих десятки и сотни тысяч объектов. В данной работе используются алгоритмы синтеза плоских представлений, имеющие субквадратичную сложность. Требование субквадратичности означает, что алгоритм уже «не имеет права» просматривать все попарные расстояния между объектами и должен обладать стратегией эффективного перебора пар. Построение такой стратегии оказывается возможным путем выявления иерархической кластерной структуры метрической конфигурации. Заодно решается проблема эффективной графической визуализации: вместо всего множества точек может отображаться только кластерная структура заданной глубины.

В данном исследовании применяется алгоритм, в котором для минимизации функционала стресса используется итерационный процесс поочередного размещения точек, состоящий из трех этапов.

- Выделение и начальное размещение S опорных точек (скелета).
- Итеративное уточнение скелета, не более I раз.
- Размещение основной массы точек относительно скелета.

На третьем этапе учитываются только расстояния между массовыми точками и точками скелета. Взаимные расстояния между массовыми точками даже не просматриваются, поэтому время размещения конфигурации из N точек имеет порядок $O(S^2 I) + O(SN)$. Оно линейно по N, если размер скелета достаточно мал и фиксирован, и квадратично, если S имеет порядок N. Отбрасывание части информации существенно ускоряет работу алгоритма, но несколько ухудшает качество размещения. Однако если скелет отражает наиболее значимые особенности исходной метрической конфигурации, то ухудшение качества размещения будет незначительным. В [5,6] показано, что если конфигурация имеет ε-кластерную структуру, то скелет содержит точки из каждого кластера.

Первый этап начинается с размещения пары точек. Берётся пара точек, расстояние R между которыми близко к максимальному. Этим точкам назначаются координаты (0,0) и (R,0). Затем к размещенным точкам присоединяется по одной точке. Очередная точка выбирается так, чтобы расстояние от нее до ближайшей размещенной было максималь-

но. При этом новые точки образуют все более и более мелкую сетку. Процесс продолжается, пока S точек не окажутся размещенными. Последней будет размещена точка, участвующая в паре с наименьшим расстоянием.

На втором этапе производится несколько «больших» итераций, в которых поочередно уточняется размещение всех скелетных точек, пока значение стресса не перестанет существенно уменьшаться. Задача второго этапа — как можно точнее выстроить скелет, так как относительно него размещается основная масса точек.

На третьем этапе размещается основная масса точек. Очерёдность их размещения не имеет значения, поскольку взаимные расстояния между ними не учитываются.

При размещении отдельных точек на всех трёх этапах используется метод Ньютона-Рафсона. Начальное приближение точки строится по ближайшим скелетным точкам.

Описанный порядок размещения точек ориентирован на сохранение наиболее существенных структурных особенностей исходной метрической конфигурации. Сначала выстраивается скелет конфигурации из небольшого количества опорных точек. Таких точек немного, поэтому скелет размещается с незначительными искажениями. По мере измельчения сетки происходит плавный переход к размещению точек в тех областях, где уже имеется достаточно много локальных соседей. При выборе $\alpha < 0$ именно ближайшие соседи дают основной вклад в функционал стресса. Поэтому начальное приближение по трем ближайшим точкам позволяет практически сразу указать расположение точки, близкое к оптимальному. Это приводит к заметному повышению эффективности, поскольку при удачном начальном расположении требуется существенно меньшее число итераций. Особенно эффективен данный метод при размещении конфигураций, изначально близких к двумерным евклидовым. Если исходная конфигурация в точности двумерна, она почти всегда размещается точно, даже при полном отключении уточняющих итераций по методу Ньютона-Рафсона.

На основе описанного базового алгоритма разработан иерархический алгоритм размещения сверхбольших метрических конфигураций. После построения скелета верхнего уровня размещается скелет второго уровня, затем скелет третьего уровня, и т.д. При этом скелет каждого уровня размещается только относительно точек скелета предыдущего уровня. Именно это и позволяет достичь субквадратичной эффективности. Одновременно строится иерархическая кластерная структура метрической конфигурации.

2.3.2 Карта сходства всех ресурсов

Ниже приведены полная карта сходства всех ресурсов и ее фрагмент, полученные с помощью алгоритма многомерного шкалирования.

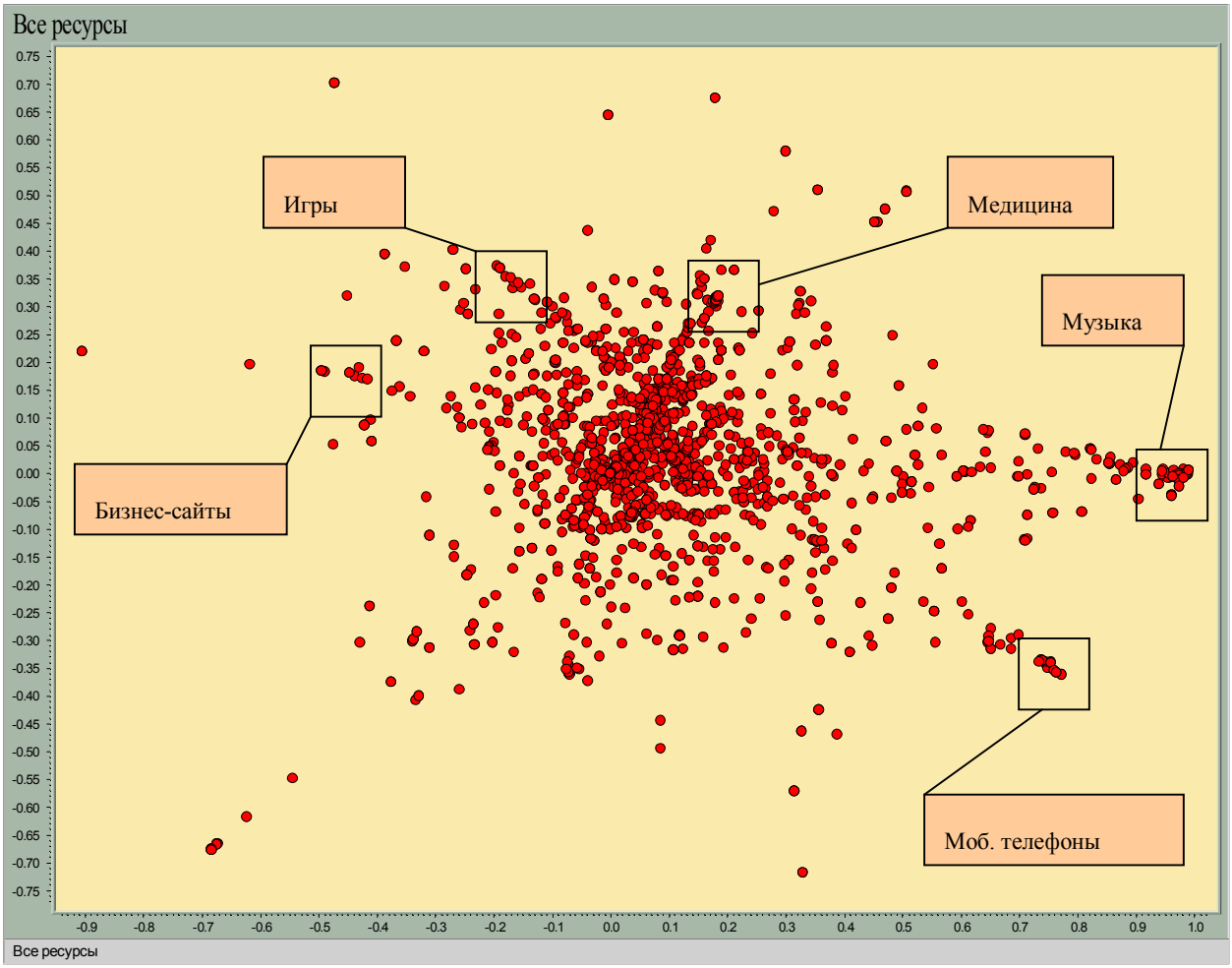


Рис.7. Полная карта сходства ресурсов.

В полученной карте сходства всех ресурсов выявились хорошо интерпретируемые группы ресурсов. Некоторые из них показаны на рисунке 7. Ресурсы, которые объединились в плотные группы, как правило, имеют схожую тематику. В частности, из плотной группы 37 сайтов, показанных на рис.8, все посвящены mp3 музыке, покупке музыкальных компакт-дисков, плееров и т.д. Лишь 4 из них напрямую не относятся к музыке, в то же время, это поисковые сайты и Интернет-магазин, имеющие возможность поиска и покупки музыки.

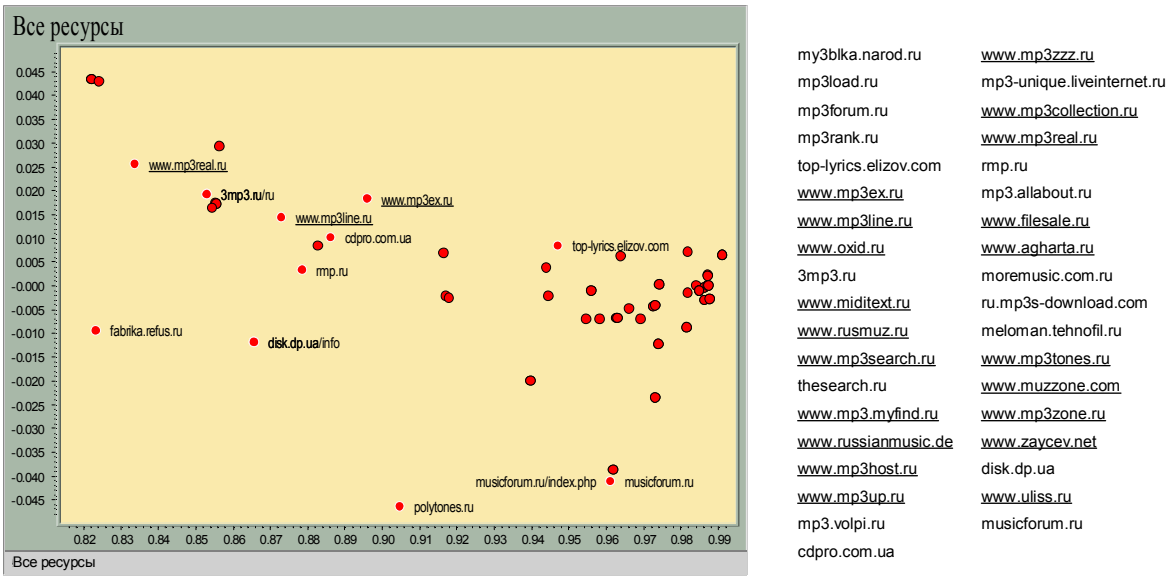
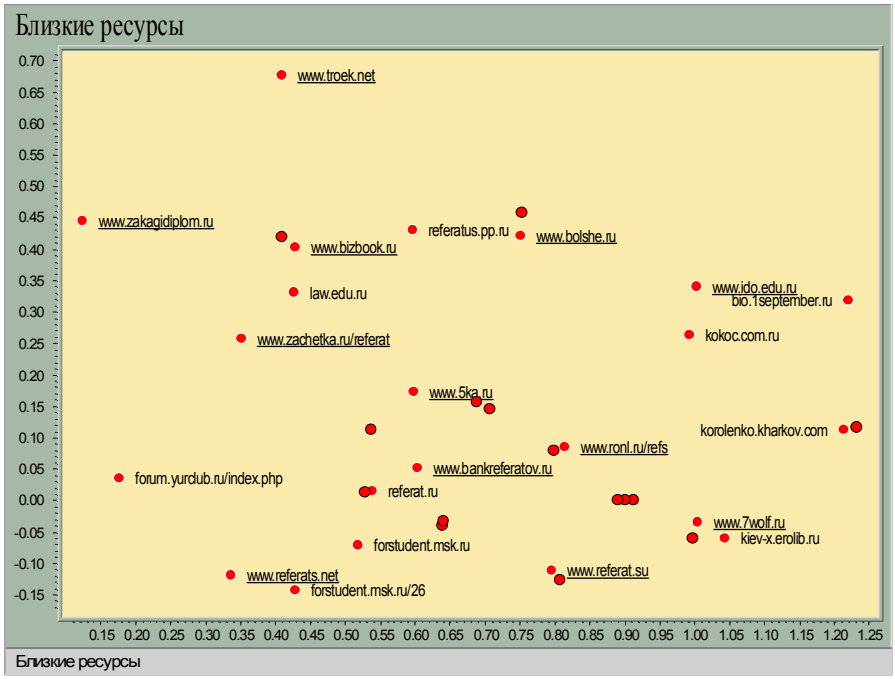


Рис.8. Фрагмент карты сходства ресурсов и список сайтов в группе «Музыка».

2.4 Построение локальной карты сходства ресурсов

От более общего случая построения карты сходства всех ресурсов перейдем к построению локальной карты сходства ресурсов, близких к некоторому заданному ресурсу. Алгоритм заключается в выделении ресурсов, расстояние до которых от заданного меньше определенного порога. Порог выбирается таким образом, чтобы либо получить необходимое число схожих ресурсов, либо увидеть все ресурсы, достаточно близкие к заданному. Пример построения списка схожих сайтов и локальная карта сходства приведены на рисунке 9. Как видно, почти все ресурсы связаны с рефератами. Преимуществом локальной карты сходства по сравнению с фрагментом общей карты (рис.8) является более точное размещение точек на карте. Положения точек на общей карте подвержены дополнительным искажениям, наведенным далекими точками.

Данный алгоритм позволяет построить область близких ресурсов не только для одного заданного ресурса, но и для произвольной группы ресурсов. Таким образом, можно предложить пользователю перечень новых или ранее не посещенных сайтов, схожих с теми, которые он уже посетил или посетили близкие к нему пользователи.



www.troek.net
www.referat.su/refs_new
referat.ru
edu.mton.ru/index.php
referatw.ru/cgi-bin/main.cgi
referat.ru/document
devir.ru/dev/info226.php
www.erudition.ru/referat/ref
law.edu.ru
www.ido.edu.ru
forstudent.msk.ru/26
www.7wolf.ru
www.bankreferatov.ru
www.zakagidiplom.ru
www.ronl.ru/refs
forum.yurclub.ru/index.php
www.bolshe.ru/book
kiev-x.erolib.ru
kiev-x.erolib.ru/miNNet.php
www.5ka.ru
www.bizbook.ru/detail.html
www.ronl.ru
www.zachetka.ru/referat
www.referat.su
korolenko.kharkov.com/m
edu.mton.ru
forstudent.msk.ru
bio.1september.ru
www.referats.net
devir.ru
referatus.pp.ru
kokoc.com.ru
korolenko.kharkov.com
www.erudition.ru/referat
www.bolshe.ru
www.bizbook.ru
devir.ru/dev

Рис.9. Карта сходства ресурсов, близких к «www.troek.net» и список этих ресурсов.

3 Заключение

3.1 Основные результаты

В итоге проделанной работы были получены следующие основные результаты.

1. Разработаны и реализованы эффективные алгоритмы анализа логов поисковой машины, приводящие в конечном итоге к построению хорошо интерпретируемых карт сходства ресурсов Интернет.

2. Предложена методика построения локальных карт сходства, позволяющая находить и визуализировать схожие по посещаемости сайты, а также генерировать для произвольного заданного пользователя адресное предложение ресурсов.

3.2 Направления дальнейших исследований

Описанная методика может быть развита и применена для ряда важных задач, исследование которых изначально не входило в рамки данной работы. Наиболее интересным направлением дальнейших исследований в этой области является вычисление взаимосогласованных оценок сходства пользователей и ресурсов. Для решения этой задачи применяется принцип взаимного согласования метрик на множествах услуг (ресурсов) и клиентов (пользователей).

Мы считаем, что клиенты схожи, если они пользуются схожими ресурсами. И, наоборот, ресурсы схожи, если ими пользуются схожие клиенты. Задав каким-либо образом метрику на ресурсах, можно посчитать матрицу сходства пользователей. Аналогично, зная метрику на пользователях, можно посчитать сходство ресурсов. Данные преобразования являются рекурсивными и позволяют организовать итерационный вычислительный процесс, в котором обе меры сходства уточняются поочередно. Судя по результатам предварительных экспериментов такого рода процессы сходятся достаточно быстро.

Этот алгоритм был также реализован и проверен на тестовых данных. На вход алгоритму подавались сгенерированные попарные расстояния точек двух классов, расположенных на плоскости. Затем сравнивались полученная карта сходства и истинное расположение точек в тестовых данных. Хорошее соответствие точек наблюдалось при большом их числе и при условии, что точки разных классов не сильно удалены друг от друга.

Другой метод, который можно применить для построения взаимосогласованных мер сходства — это оценка расстояний с помощью неравенств треугольника. При этом предполагается, что задана евклидова метрика между пользователями и ресурсами, тогда все недостающие расстояния будут оценены сверху и снизу с помощью неравенств треугольника. Дальнейшее построение расстояний заключается в итерационном процессе нахождения минимального интервала, выбором в качестве расстояния между соответствующими точками середины этого интервала, и новом пересчете (коррекции) остальных интервалов и т.д.

Метод, основанный на оценке расстояний с помощью неравенств треугольника также был опробован на тестовых данных и дал хорошее соответствие с исходной конфигурацией для случая точек на плоскости. К сожалению, этот метод обладает слишком низкой эффективностью, т.к. требует перебора всевозможных троек точек. Кроме того, он плохо работает на сильно разряженных матрицах, с элементами, редко отличающимися от 0 и 1.

Интересным направлением дальнейших исследований является также разработка алгоритмов построения ранжированного списка ресурсов, рекомендуемых данному пользователю. Решение данной задачи может быть использовано различными способами.
Во-первых, для персонализации результатов поиска, когда каждому пользователю предоставляется список ресурсов, ранжированный согласно его личным предпочтениям.

Во-вторых, для генерации персонального предложения — списка или карты ресурсов, которые с большой вероятностью могут заинтересовать данного пользователя, поскольку они близки к ресурсам, которые посещал данный пользователь, или схожие с ним пользователи.

Также интересна задача построения ранжированного списка рекомендуемых ресурсов, близких данному ресурсу. Решение данной задачи может быть использовано для предоставления новой услуги владельцам ресурсов, когда они получают возможность разместить на своем сайте карту близких ресурсов, популярных среди пользователей данного ресурса. Эту карту можно включать, например, в традиционный раздел «Links». Кроме того, построение ранжированного списка рекомендуемых ресурсов, по сути дела, является новым способом представления результатов поиска.

Список литературы

[1] Ternary Seach Trees, Dr. Dobb's Journal April 1998 // http://www.ddj.com/documents/s=921/ddj9804a/

[2] Mastering Algorithms with C. By Kyle Loudon, First Edition August 1999, Chapter 8: Hash Tables // http://www.oreilly.com/catalog/masteralgoc/chapter/ch08.pdf

[3] Represent sparse matrices by some appropriate form of linked lists. By Tran Van Canh // http://www.codeproject.com/cpp/sparse_matrices.asp

[4] М.Дэйвисон. Многомерное шкалирование. Методы наглядного представления данных. Перевод с английского В.С.Каменского. - Москва, «Финансы и статистика» 1988 (Multidimensional scaling Mark L.Davison. University of Minnesota).

[5] Вальков А.С. О быстрых алгоритмах синтеза плоских представлений конечных метрических конфигураций // Ж. вычисл. матем. и матем. физ. 2005. Т.45. №2, с.357-368.

[6] Вальков А.С. О быстром алгоритме восстановления иерархической ε-кластерной структуры при $\varepsilon < 1$ // Ж. вычисл. матем. и матем. физ. 2005. Т.45. №1, с.170-179.