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

Технология персонализации на основе выявления тематических профилей пользователей и ресурсов Интернет

Лексин Василий Алексеевич

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА МАГИСТРА

Магистерская программа 511656 «Математические и информационные технологии»

Научный руководитель:
н.с. ВЦ РАН, к.ф.-м.н. К. В. Воронцов

Москва 2007 г.

Содержание

1. ВВЕДЕНИЕ 3

1.1 Постановка задачи 5

1.2 Краткое содержание работы по главам 6

2. ОБЗОР ЛИТЕРАТУРЫ 7

2.1 Ко-кластеризация 7

2.1.1 Коллаборативная фильтрация с помощью ко-кластеризации (co-clustering) 7

2.1.2 Алгоритм Бергмана: 8

3. МЕТОДЫ ОЦЕНИВАНИЯ СХОДСТВА ПОЛЬЗОВАТЕЛЕЙ И РЕСУРСОВ 9

3.1 Точный тест Фишера 9

3.2 Вероятностная постановка задачи 10

3.2.1 EM-алгоритм 10

3.2.2 Коллаборативная фильтрация на основе EM-алгоритма 12

3.2.3 Описание алгоритма 15

3.3 Идея построения обобщенного профиля 17

4. ВЫЧИСЛИТЕЛЬНЫЕ ЭКСПЕРИМЕНТЫ 18

4.1 Проверка алгоритма на модельных данных 18

4.2 Эксперимент на данных поисковой машины 20

5. ЗАКЛЮЧЕНИЕ 24

СПИСОК ЛИТЕРАТУРЫ 24

1. Введение

В последнее время всё чаще встречается ситуация, когда компании подробно протоколируют действия своих клиентов. Актуальной проблемой становится создание новых информационных технологий для эффективного извлечения полезной и нетривиальной информации из огромного объема сырых данных о поведении клиентов.

В данной работе рассматривается несколько вариантов технологии, которая представляет собой цепочку процедур обработки данных, ведущую от исходного протокола действий клиентов к решению широкого спектра задач принятия решений, маркетинга и управления взаимоотношениями с клиентами (CRM).

Рассмотрим часто встречающуюся ситуацию, когда есть достаточно большое множество клиентов (пользователей), которые пользуются некоторым множеством услуг или ресурсов. Причем все действия клиентов протоколируются в электронном виде. Например:

- *поисковые машины* накапливают информацию о том, какие запросы делали пользователи, и какие документы они потом выбирали;
- *счетчики посещений* накапливают информацию о том, какие пользователи и когда посещали страницы, на которых установлен счетчик;
- *интернет-магазины* накапливают информацию о том, какие товары покупал каждый пользователь,
- *Интернет-провайдеры и прокси-серверы* имеют возможность собирать информацию о том, какие сайты посещал каждый пользователь, выходящий через них в Интернет,
- *веблоги и форумы* накапливают информацию о том, в каких тематических разделах принимает участие каждый пользователь, и насколько активно;
- пользователи Интернет-магазинов выбирают товары, абоненты мобильной связи пользуются разного типа услугами, и так далее.

Проблема заключается в том, чтобы научиться извлекать некоторую полезную информацию из этого огромного объема сырых данных, необходимую для решения ряда аналитических задач по персонализации контента, прогнозированию, выявлению предпочтений пользователей, выделению групп схожих ресурсов и т.д. Перечислим некоторые примеры таких аналитических задач:

Кластеризация ресурсов — группирование схожих по множеству посетителей ресурсов в несколько кластеров (групп) ресурсов. Кластеризация позволяет строить каталоги ресурсов, а также выявлять недостатки существующих тематических каталогов.

Кластеризация пользователей — группирование схожих пользователей в кластеры аналогично кластеризации ресурсов. Позволяет выявлять группы пользователей со схожими интересами.

Построение устойчивых поведенческих профилей пользователей в виде перечня групп ресурсов, посещаемых как данным пользователем, так и схожими с ним пользователями.

Построение расширенных профилей пользователей, включающих социально-демографические данные (анкеты), описательные статистики и поведенческие профили. Расширенные профили позволяют классифицировать новых пользователей, выявлять зависимости между пользовательским поведением и социально-демографическими характеристиками.

Сегментация клиентской базы на основе расширенных профилей позволяет выделять сегменты как по анкетным данным клиентов, так и по их поведению. Эта информация используется при маркетинговых исследованиях.

Прямой маркетинг — предоставление рекламы и маркетинговых предложений конкретному пользователю на основе его поведенческого профиля.

Персонализация контента — представление каждому пользователю сайта наиболее интересной для него информации в наиболее удобном для него виде. Знание информационных предпочтений пользователя позволяет динамически перестраивать контент сайта.

Построение Карт Сходства ресурсов и пользователей — отображение множества наиболее посещаемых ресурсов и наиболее активных пользователей в виде точечного графика. Схожим ресурсам (пользователям) соответствуют близкие точки на карте. Карту Сходства можно использовать как графическое средство навигации.

Существует множество методов и подходов к решению поставленных выше задач. Весь класс этих методов принято называть методами *коллаборативной фильтрации*.

Для решения задач коллаборативной фильтрации в компании Форексис разработана технология Анализа Клиентских Сред (АКС) [8], основанная на понятии сходства. **Клиенты схожи, если они пользуются схожим набором ресурсов. Ресурсы схожи, если ими пользуются схожие клиенты.** Этот интуитивно очевидный принцип сходства можно применить для построения мер сходства, или метрик, как на множестве клиентов, так и на множестве ресурсов. Данная работа посвящена рассмотрению нескольких методов в рамках АКС.

Целью работы является разработка, реализация, экспериментальная проверка и сравнение различных алгоритмов оценивания сходства пользователей и ресурсов. Первый алгоритм основан на оценке неслучайности совместных посещений. В данной работе предлагается использовать для этого точный тест Фишера (гипергеометрическое распределение). Второй алгоритм основан на идее выявления скрытых интересов пользователей и темы ресурсов по исходным данным о посещениях. Алгоритм строит для каждого пользователя и каждого ресурса так называемый тематический профиль, который представляет собой вектор оценок, соответствующих некоторым темам. Профиль пользователя характеризует степень его заинтересованности каждой из тем. Профиль ресурса характеризует способность данного ресурса удовлетворять заинтересованных пользователей по этому набору тем. Количество тем (длина профиля) и сами темы могут либо задаваться априори, либо определяться алгоритмом автоматически. Для сравнения рассмотрим третий алгоритм, основанный на идее ко-кластеризации—одновременной кластеризации множества ресурсов и пользователей.

Актуальность работы очевидна в связи с растущей популярностью идеи персонализации информационных услуг, а также в связи с развитием CRM (систем управления взаимодействия с клиентами).

Новым в данной работе, по сравнению с ранее реализованными методами АКС, является применение специализированного итерационного алгоритма, позволяющего оценивать профили пользователей и ресурсов, существенно сокращать объемы хранимых в памяти данных, строить метрики на множествах пользователей и ресурсов, получать визуальные представления множеств пользователей и ресурсов в виде карт сходства, эффективно решать задачи персонализации.

В рамках исследования проведены вычислительные эксперименты на модельных данных, а также на недельном периоде данных поисковой системы.

Показано, что алгоритм, основанный на выявлении тематических профилей, дает минимальную погрешность на модельных данных и реальных данных и хорошо интерпретируемую карту сходства ресурсов Интернет.

1.1 Постановка задачи

Исходными данными являются протоколы действий пользователей. Каждая запись протокола описывает событие «пользователь *u* выбрал ресурс *r*». В зависимости от конкретной задачи (предметной области) запись может содержать следующую информацию: идентификатор пользователя, название ресурса, тип, время начала и продолжительность действия и т.д. Например, действием может быть посещение страницы, выбор товара или услуги в интернет-магазине, проставление рейтинга просмотренного фильма и т. д.

Введем некоторые понятия, необходимые для формальной постановки задачи.

Пусть есть R — множество ресурсов,

U — множество пользователей.

Задан протокол пользования $D = (u_i, r_i)$, $i = 1,...,l$. То есть множество событий типа пользователь $u \in U$ использовал ресурс $r \in R$, l— число записей в протоколе.

По нему строится матрица пользования $A_{|U| \times |R|}$. В зависимости от конкретной задачи A_{ur} могут нести различную информацию о пользовательском поведении. Это может быть бинарная информация о посещении или не посещении заданного ресурса данным пользователем, частота (или число) пользований ресурса r пользователем u, стоимость или рейтинг, проставленный пользователем *u* для ресурса *r* и т.д.

Матрица $A_{|U| \times |R|}$ чаще всего бывает сильно разреженной (далеко не все пользователи используют все ресурсы и, наоборот, далеко не всеми ресурсами пользуются все пользователи). Поэтому множество посещений нам часто удобнее хранить не в виде громоздкой матрицы, а в виде набора непустых ее элементов: $D_A = \{(u,r) \mid A_{ur} > 0\}$. D_u и D_r — множества непустых элементов в матрице посещений при фиксированном *u* и при фиксированном *r* соответственно.

Задача заключается в том, чтобы по имеющимся данным построить функции сходства (метрики) на множестве пользователей $\rho_U : U \times U \to R_+$ и на множестве ресурсов $\rho_R : R \times R \to R_+$ таким образом, чтобы близкими по метрике ρ_U были пользователи, которые пользуются схожими ресурсами, и близкими по метрике ρ_R были ресурсы, которыми пользуются схожие клиенты.

В данной работе рассматривается подход, при котором сначала по исходным данным вычисляются векторные описания ресурсов и пользователей, называемые в работе профилями. Затем функции сходства определяются как евклидовы метрики над профилями. Причем компоненты профилей интерпретируются как тематики, которыми могут интересоваться пользователи и которые могут удовлетворять (в той или иной степени) ресурсы.

Пусть T — множество тем в профиле.

Тематический профиль ресурса *r* и пользователя *u* запишем следующим образом:

$P_r = (p_{1r}, p_{2r}, ..., p_{|T|r})$

$P_u = (p_{1u}, p_{2u}, ..., p_{|T|u})$.

Подчеркнем, что природа профилей пользователей и ресурсов одинакова, поэтому, зная профили для всех ресурсов и пользователей, мы можем оценить расстояния (например, как среднеквадратичное отклонение) как от ресурса до ресурса и от пользователя до пользователя, так и от пользователя до ресурса:

$$\rho_R(r, r') = \rho(P_r, P_{r'}) = \sqrt{\sum_{t=1}^{|T|} (p_{1r} - p_{1r'})^2} .$$

Аналогично

$$\rho_U(u, u') = \rho(P_u, P_{u'}) = \sqrt{\sum_{t=1}^{|T|} (p_{1u} - p_{1u'})^2}$$

и

$$\rho(r, u) = \rho(P_r, P_u) = \sqrt{\sum_{t=1}^{|T|} (p_{1r} - p_{1u})^2} .$$

Таким образом, мы можем построить метрику на множествах ресурсов и пользователей. В качестве критерия качества построенной метрики применяется метод *k* ближайших соседей (kNN) Качество построения оценивается по функционалу числа ошибок при попытке классифицировать точки методом kNN, используя частичную классификацию ресурсов. В данной работе рассматривается несколько алгоритмов оценивания расстояний, качество работы алгоритмов оценивается по данному функционалу.

1.2 Краткое содержание работы по главам

Работа состоит из введения, трёх глав и заключения.

Во введении обсуждается круг проблем, возникающих в области извлечения полезной информации из огромного объема сырых данных о пользовательском поведении. Вводятся основные определения и обозначения, ставится формальная постановка задачи и излагаются основные идеи для решения поставленных задач.

В первой главе описывается один из стандартных алгоритмов решения задачи коллаборативной фильтрации, основанный на совместной кластеризации пользователей и ресурсов.

Во второй главе сначала описывается алгоритм построения меры сходства на множестве ресурсов с помощью точного теста Фишера. Далее описывается алгоритм EM и алгоритм, основанный на построении профилей пользователей и ресурсов, который вбирает в себя идеи из EM алгоритма и является основным предметом рассмотрения в данной работе. В заключение главы описывается идея построения обобщенных профилей пользователей и ресурсов.

Третья глава посвящена описанию экспериментальных результатов и сравнению алгоритмов. Сначала описывается работа алгоритма построения профилей на модельных данных, затем на данных поисковой машины. Далее сравнивается работа двух алгоритмов построения мер сходства на множестве ресурсов и приводятся соответствующие карты сходства ресурсов.

В заключении указываются основные результаты и перспективы дальнейших исследований в данной области.

2. Обзор литературы

Рассмотрим один из алгоритмов решения задачи коллаборативной фильтрации, который достаточно популярен, часто используются в прикладных задачах и описан в литературе.

2.1 Ко-кластеризация

Алгоритм ко-кластеризации позволяет вычислить профили пользователей и ресурсов. Основной идеей алгоритма является одновременное получение групп (кластеров) ближайших пользователей и ресурсов, основываясь на заданной матрице пользования A.

Пусть $W_{|U|\times|R|}$ — матрица доверия значениям в матрице A. $W_{ur} \in [0,1]$, где $u \in U$, $r \in R$. Причем $W_{ur} = 1$ в случае полного доверия соответствующему значению, $W_{ur} = 0$ при полном ему недоверии.

Необходимо построить алгоритм совместной кластеризации пользователей и ресурсов, т.е. пару функций $\rho : U \mapsto \{1,...,k\}$ и $\gamma : R \mapsto \{1,...,l\}$, которые каждому пользователю и каждому ресурсу ставят в соответствие номер кластера, k и l — количество кластеров пользователей и ресурсов соответственно. Пара (ρ, γ) определяет соответствующий ко-кластер.

Для выполнения ко-кластеризации решается оптимизационная задача

$$\sum_{(u,r)\in T} W_{ur}\left(A_{ur} - \text{Ā}_{ur}\right)^2 \to \min_{(\rho,\gamma)}, \tag{1}$$

где Ā_{ur} — оценка среднего по ко-кластеру $(p(u), \gamma(r))$.

Существует несколько методов и подходов вычисления оценок Ā_{ur}. Простейший: Ā_{ur} — среднее значение в соответствующем ко-кластере. В работе [5] предложена более сложная оценка, которая учитывает предпочтения каждого пользователя и особенности каждого ресурса, путем введения дополнительных членов:

$$\text{Ā}_{ur} = A_{gh}^{COC} + (A_u^R - A_g^{RC}) + (A_r^C - A_h^{CC}) \tag{2}$$

где $g = \rho(u)$, $h = \gamma(r)$, A_u^R, A_r^C — средние значения пользователя u и ресурса r; A_{gh}^{COC}, A_g^{RC} и A_h^{CC} — средние значения соответствующих ко-кластера, кластера пользователя и кластера ресурса.

2.1.1 Коллаборативная фильтрация с помощью ко-кластеризации (co-clustering)

Для решения задачи (1) известен алгоритм ко-кластеризации Бергмана. Он оценивает средние значения A_{gh}^{COC}, A_g^{RC}, A_h^{CC}, A_u^R и A_r^C для каждого ко-кластера (ρ, γ). Ключевая идея алгоритма заключается в том, чтобы задать некоторую начальную ко-кластеризацию и затем поочередно оптимизировать по строке (пользователю) и по столбцу (ресурсу) кластеризацию, пока не будет достигнута сходимость. Ниже приведены основные шаги алгоритма.

2.1.2 Алгоритм Бергмана:

Вход: Матрица пользования *A*, ненулевая матрица *W*, число кластеров пользователей *l*, число кластеров ресурсов *k*.

Выход: Локально-оптимальное разделение на ко-кластеры (ρ, γ) и матрицы средних значений A_{gh}^{COC}, A_g^{RC}, A_h^{CC}, A_u^R и A_r^C.

Алгоритм:

1. Случайным образом инициализировать ко-кластеры (ρ, γ)

2. повторять

а) Вычислить средние $A_{gh}^{COC}, A_g^{RC}, A_h^{CC}, A_i^R u A_j^C$

б) Обновить кластеры по строкам

$$\rho(u) = \arg\min_{1 \le g \le k} \sum_{r=1}^{n} W_{ur} (A_{ur} - A_{g\gamma(r)}^{COC} - A_u^R + A_g^{RC} - A_r^C - A_{\gamma(r)}^{CC})^2, 1 \le u \le m$$

в) Обновить кластеры по столбцам

$$\gamma(r) = \arg\min_{1 \le h \le l} \sum_{u=1}^{m} W_{ur} (A_{ur} - A_{\rho(u)h}^{COC} - A_u^R + A_{\rho(u)}^{RC} - A_r^C - A_h^{CC})^2, 1 \le r \le n$$

пока не сойдется.

В [5] получены оценки времени работы алгоритма. Время формирования кластеров по строкам и по столбцам занимает $O(mkl)$ и $O(nkl)$ соответственно. Общее число итераций во всем вычислении $O(W^{glob} + mkl + nkl)$, где W^{glob} – число ненулевых элементов в A.

Данный алгоритм позволяет заполнить пропущенные значения в *A*. Эта особенность алгоритма позволяет применять его для предсказания рейтингов в рекомендательных системах. Необходимый рейтинг для заданных ресурса и пользователя мы можем определить по формуле (2). Когда только пользователь (ресурс) является новым, то предсказанное значение берем как среднее по кластеру пользователя (ресурса). А когда и ресурс и пользователь являются новыми, предсказанное значение – средний рейтинг по всей матрице.

Еще одно достоинство алгоритма Бергмана — его можно приспособить для решения динамической задачи. В динамическом случае матрица рейтингов *A* изменяется во времени, и поэтому мы требуем, чтобы матрица оценок $\hat{A}$ приспосабливалась к изменениям в матрице *A*, которая будет дополняться новыми значениями по мере появления новых пользователей и ресурсов. Для решения этой задачи рассмотрим механизм наращиваемого обновления. Так как значения $\hat{A}_{ur}$ зависят только от средних значений (статистик) по кластерам, то по мере появления новых ресурсов и пользователей нам необходимо каким-то образом обновлять эти статистики. Когда вновь появившееся значение соответствует новому пользователю или ресурсу и еще не определен кластер для этого нового ресурса или пользователя, то мы временно определяем этот новый ресурс или пользователя к временному переходному кластеру до тех пор, пока мы не обновим средние значения кластеров. При следующем проходе алгоритма пользователи и ресурсы в переходном кластере переопределяются в один из постоянных кластеров.

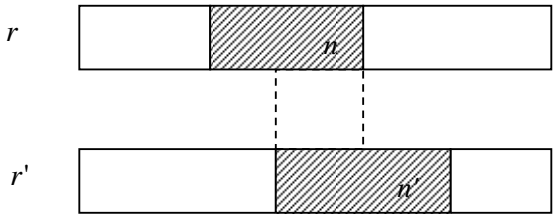
3. Методы оценивания сходства пользователей и ресурсов

Рассмотрим два различных подхода оценивания сходства пользователей и ресурсов. Один из них основан на проверке статистической гипотезы о независимости посещения, а второй на построении тематических профилей пользователей и ресурсов.

3.1 Точный тест Фишера

Сформулируем задачу вычисления оценок сходства ресурсов. Пусть ресурсы $r, r' \in R$ посещались n и n' пользователями соответственно. Пусть $n_{rr'}$ пользователей посетили оба ресурса. Если предположить, что посещения ресурсов r и r' являются независимыми событиями, то количество чисто случайных посещений обоих ресурсов одним и тем же пользователем будет подчиняться гипергеометрическому распределению:

$$P_{rr'} = P(n_{rr'} = x) = \frac{C_n^x C_{|U|-n}^{n'-x}}{C_{|U|}^{n'}}.$$



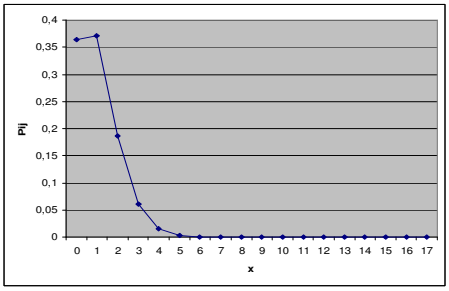


Рис.1. Построение оценки сходства двух ресурсов. График гипергеометрического распределения при $|U| = 104$, $n = 100$, $n' = 100$.

Эта вероятность максимальна при $x \approx \frac{nn'}{|U|}$ и быстро убывает по мере увеличения x. Если $n_{rr'}$ настолько велико, что $P_{rr'} < \alpha$ при заданном достаточно малом уровне значимости α, то приходится признать, что либо реализовалось маловероятное событие, либо исходная гипотеза о независимости ресурсов неверна, следовательно имеется статистически значимая взаимосвязь в посещениях данной пары ресурсов, следовательно они близки.

Если же $P_{rr'} > \alpha$, то наблюдаемое распределение посещений (n, n', $n_{rr'}$) вполне могло реализоваться случайно, и делать какие-либо выводы о сходстве ресурсов r и r' невозможно.

Введем множество пар ресурсов $R = \{(r, r') \mid P_{rr'} < \alpha\}$. Чем меньше вероятность $P_{rr'}$, тем более схожи ресурсы. Расстояние между ресурсами будем оценивать по формуле $\rho(r, r') = M(P_{rr'})$, где M — некоторая монотонно возрастающая функция. Таким образом, функция расстояния определяется только на подмножестве пар R, 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 $\rho(r,r') = \left(\frac{|\ln \alpha|}{|\ln P_{rr'}|} \right)^3$, где α — уровень значимости. Видно, что функция $\rho(r,r')$ лежит в интервале [0,1], является монотонно возрастающей функцией $P_{rr'}$ и имеет максимум при $P_{rr'} = \alpha$, равный единице.

Этот подход рассмотрен в работах [7,10,11]. В данной работе предлагается принципиально иной способ оценивания расстояния между ресурсами, также основанный на вероятностной постановке задачи.

3.2 Вероятностная постановка задачи

Допустим, что у каждого пользователя $u \in U$ имеется некоторое множество интересов или потребностей, которые мы будем называть темами. Множество всех возможных тем обозначим через *T*. Допустим, пользователь *u* имеет интерес $t \in T$ с вероятностью $p(t \mid u)$. В свою очередь, каждый ресурс соответствуем некоторому множеству тем. Допустим, ресурс *r* удовлетворяет теме *t* с вероятностью *p(t|r).*

Профиль пользователя *u* определим как вектор значений вероятностей $p(t \mid u)$, $t = 1,...,|T|$, причем $\sum_{t \in T} p(t \mid u) = 1$.

Аналогично, профиль ресурса *r* — это вектор вероятностей $p(t \mid r)$, $t = 1,...,|T|$, причем $\sum_{t \in T} p(t \mid r) = 1$.

Обозначим через *p(u,r)* — вероятность выбора ресурса *r* пользователем *u.*

Метод восстановления профилей, предлагаемый в данной работе опирается на EM-алгоритм разделения смеси распределений, поэтому рассмотрим его более подробно.

3.2.1 EM-алгоритм

Пусть X — множество объектов с заданной на нем вероятностной мерой P.

Допустим, что плотность распределения на *X* имеет вид смеси *k* распределений:

$$p(x) = \sum_{j=1}^{k} w_j p_j(x), \sum_{j=1}^{k} w_j = 1,$$

где $p_j(x)$ — функция правдоподобия *j*-й компоненты смеси, w_j — её априорная вероятность. Обычно предполагается, что функции правдоподобия принадлежат заданному параметрическому семейству распределений и отличаются только значениями параметра, $p(x;\theta_j)$. Задача разделения смеси заключается в том, чтобы, зная выборку объектов $X_m = \{x_1,...,x_m\} \subset X$ и число *k* оценить вектор параметров $\Theta = (w_1,...,w_k,\theta_1,...,\theta_k)$.

Идея алгоритма заключается в следующем. Искусственно вводится вспомогательный вектор скрытых переменных G, обладающий двумя замечательными свойствами. С одной стороны, он может быть вычислен, если известны значения вектора параметров θ. С другой стороны, решение задачи максимизации правдоподобия сильно упрощается, если известны значения скрытых переменных.

EM-алгоритм состоит из итерационного повторения двух шагов. На E-шаге вычисляется ожидаемое значение вектора скрытых переменных *G* по текущему приближению вектора параметров θ. На M-шаге решается задача максимизации

правдоподобия и находится следующее приближение вектора θ по текущим значениям векторов G и θ.

Общая идея EM-алгоритма:

1: Вычислить начальное приближение вектора параметров θ;

2: повторять

3: G := EStep(θ);

4: θ := MStep(θ,G);

5: пока θ и G не стабилизируются.

Запишем принцип максимума правдоподобия:

$$\ln L(X^m) = \ln \prod_{i=1}^{m} p(x_i) = \sum_{i=1}^{m} \ln \left(\sum_{j=1}^{k} w_j p_j(x_i, \theta_j) \right) \to \max_{w_j \theta_j : \sum_{j=1}^{k} w_j = 1}$$

E-шаг (expectation). Обозначим через $p(x \& \theta_j)$ совместную плотность вероятности того, что получен объект x и этот объект сгенерирован j-й компонентой смеси. По формуле условной вероятности

$$p(x \& \theta_j) = p(x) P(\theta_j \mid x) = w_j p_j(x \mid \theta_j).$$

Введём обозначение $g_{ij} \equiv P(\theta_j \mid x_i)$. Это апостериорная вероятность того, что обучающий объект x_i был сгенерирован j-й компонентой смеси. Возьмём эти величины в качестве скрытых переменных. Обозначим $G = (g_{ij})_{m \times k} = (g_1, ..., g_j)$, где g_j — j-й столбец матрицы G. Предполагается, что каждый объект может быть сгенерирован одной и только одной компонентной. Тогда, по формуле полной вероятности

$$\sum_{j=1}^{k} g_{ij} = 1 \text{ для всех } i.$$

Зная параметры компонент w_j, θ_j, легко вычислить g_{ij} по формуле Байеса:

$$g_{ij} = \frac{w_j p_j(x_i)}{\sum_{s=1}^{k} w_s p_s(x_i)} \text{ для всех } i, j.$$

В этом и заключается E-шаг алгоритма EM.

M-шаг(maximization). Покажем, что знание значений скрытых переменных g_{ij} и принцип максимума правдоподобия приводят к оптимизационной задаче, допускающей эффективное решение.

Принцип максимума правдоподобия на M-шаге приводит к следующей постановке задачи:

$$L(X^m, \Theta) = \sum_{i=1}^{m} \ln \left(\sum_{j=1}^{k} w_j p_j(x_i, \theta_j) \right) - \lambda \left(\sum_{j=1}^{k} w_j - 1 \right) \to \max_{\Theta}.$$

Задача расщепляется на две независимые подзадачи:

1) Найдем веса w_j следующим образом:

$$\frac{\partial L}{\partial w_j} = \sum_{i=1}^{m} \frac{p_j(x_i, \theta_j)}{\sum_{s=1}^{k} w_s p_s(x_i, \theta_s)} - \lambda = 0 .$$

Помножим обе части на w_j и просуммируем по *k*:

$$\sum_{j=1}^{k} \sum_{i=1}^{m} \frac{w_j p_j(x_i, \theta_j)}{\sum_{s=1}^{k} w_s p_s(x_i, \theta_s)} = \lambda \sum_{j=1}^{k} w_j \text{ .Следовательно, } \lambda = m .$$

Для весов w_j получим:

$$w_j = \frac{1}{m} \sum_{i=1}^{m} \frac{w_j p_j(x_i \theta_j)}{\sum_{s=1}^{k} w_s p_s(x_i, \theta_s)} = \frac{1}{m} \sum_{i=1}^{m} g_{ij} .$$

2) Теперь выпишем уравнения для θ_j через g_{ij}.

$$\frac{\partial L}{\partial \theta_j} = \sum_{i=1}^{m} \frac{w_j \frac{\partial p_j(x_i, \theta_j)}{\partial \theta_j}}{\sum_{s=1}^{k} w_s p_s(x_i, \theta_s)} = \sum_{i=1}^{m} \frac{w_j p_j(x_i, \theta_j) \frac{\partial}{\partial \theta_j} \ln p_j(x_i, \theta_j)}{\sum_{s=1}^{k} w_s p_s(x_i, \theta_s)} =$$

$$= \sum_{i=1}^{m} g_{ij} \frac{\partial}{\partial \theta_j} \ln p_j(x_i, \theta_j) = \frac{\partial}{\partial \theta} \sum_{i=1}^{m} g_{ij} \ln p_j(x_i, \theta_j) = 0, \; j = 1,...,k .$$

Таким образом, приходим к *k* независимым задачам максимизации взвешенного правдоподобия для определения вектора параметров θ_j:

$$\sum_{i=1}^{m} g_{ij} \ln p_j(x_i, \theta_j) \to \max_{\theta_j} \tag{3}$$

Здесь объекты выборки учитываются с весами g_{ij}, причём распределение весов — своё для каждой из *k* компонент.

3.2.2 Коллаборативная фильтрация на основе EM-алгоритма

Распишем вероятность выбора ресурса *r* пользователем *u*:

$$p(u, r) = \sum_{t} p(u)\, p(t \mid u) q(r \mid t, u) ,$$

где $p(u) \equiv p_u$ — априорная вероятность того, что выбор будет сделан пользователем *u*, $\sum_{u \in U} p_u = 1$;

$p(t \mid u) = p_{tu}$, $\sum_{t \in T} p_{tu} = 1$ для всех $u \in U$ — вероятность того, что пользователь *u* в данный момент интересуется темой *t*;

$q(r \mid t, u) = q(r \mid t)$ — апостериорная вероятность того, что будет выбран ресурс *r* при условии, что выбор делает пользователь *u*, интересующийся темой *t*.

Таким образом, вероятность $p(u,r)$ будем записывать как сумму произведений этих трех вероятностей по всем темам.

Апостериорную вероятность $q(r \mid t)$ найдем по формуле Байеса:

$$q(r \mid t) = \frac{q_{tr} q_r}{\sum_{s \in R} q_{ts} q_s}.$$

Подставим это выражение в формулу для $p(u,r)$:

$$p(u \mid r) = \sum_t p_u q_r p_{tu} \frac{q_{tr}}{\sum_{s \in R} q_{ts} q_s} \tag{4}$$

Приведенные выше формулы аналогичным образом можно записать относительно ресурсов*:*

$$p(u,r) = \sum_{t \in T} q(r) q(t \mid r) p(u \mid t,r),$$

где $q(r) = q_r$ — значения априорной вероятности выбора ресурса *r;*

$q(t \mid r) = q_{tr}$ — вероятность того, что ресурсу r соответствует тема *t*;

$p(u \mid t,r) = p(u \mid t)$ — апостериорная вероятность того, что пользователь *u* интересуется темой *t,* зайдя на ресурс *r*. Здесь и в предыдущем случае мы опираемся на гипотезу независимости посещения пользователем различных ресурсов и считаем, что $p(u \mid t,r)$ не зависит от *r*.

Таким образом, мы можем записать формулу для вероятности $p(u \mid r)$ через профили ресурсов:

$$p(u,r) = \sum_{t \in T} p_u q_r q_{tr} \frac{p_{tu}}{\sum_{s \in R} p_{ts} p_s} \tag{5}$$

Воспользуемся принципом максимума правдоподобия для выборки посещений $D = (u_i, r_i)_{i=1}^l$:

$$\ln \prod_{i=1}^{l} p(u_i, r_i) \to \max_{p_{tu}, q_{tr}} \tag{6}$$

Основная идея алгоритма восстановления профилей p_{tu} и q_{tr} по выборке *D* заключается в последовательном чередовании двух действий:

1) оптимизировать p_{tu} при фиксированном q_{tr},
2) оптимизировать q_{tr} при фиксированном p_{tu}

и так далее в итерациях пока не будет достигнута сходимость.

Рассмотрим задачу максимума правдоподобия (МП) для формулы (4).

Запишем Лагранжиан:

$$L(p_{tu}) = \sum_{(u,r) \in D} \ln p_u \sum_{t \in T} p_{tu} \frac{q_{tr} q_r}{\sum_s q_{ts} q_s} - \sum_u \lambda_u \left(\sum_{t \in T} p_{tu} - 1 \right) \to \max_{p_{tu}}$$

При ограничениях:

$\sum_t p_{tu} = 1, \forall u \in U$;

$p_{tu} \geq 0, \forall t \in T, \forall u \in U$.

Продифференцируем по p_{tu} и прировняем к нулю:

$\frac{\partial L}{\partial p_{tu}} = \sum_{r \in D_u} \frac{1}{p_{tu}} \frac{p_{tu} q(r \mid t)}{\sum_{s \in T} p_{su} q(r \mid s)} - \lambda_u = 0$. Здесь мы умножили и разделили на p_{tu} .

Введем обозначение для так называемых скрытых компонент:

$$H_{tr}(u) = \frac{p_{tu} q(r \mid t)}{\sum_{s \in T} p_{su} q(r \mid s)}, \qquad (7)$$

и заметим, что это выражение есть ни что иное, как формула Байеса. Можно дать вероятностную интерпретацию данных компонент следующим образом: вероятность того, что пользователь *u*, зайдя на ресурс *r,* интересовался темой *t*.

Тогда

$\sum_{t \in T} H_{tr}(u) = 1$.

Далее разнесем два члена уравнения в разные части равенства, помножим обе части на p_{tu} и просуммируем по t:

$\sum_{t \in T} \sum_{r \in D_u} \frac{p_{tu} q(r \mid t)}{\sum_s p_{su} q(r \mid s)} = \sum_{t \in T} \lambda_u p_{tu}$.

Учитывая, что $\sum_{t \in T} p_{tu} = 1$ и $\mid D_u \mid = p_u \mid R \mid$, получаем:

$\sum_{r \in D_u} 1 = \lambda_u$, следовательно $\lambda_u = \mid R \mid p_u$.

Таким образом, получаем

$$p_{tu} = \frac{\sum_{r \in D_u} H_{tr}(u)}{\sum_{r \in D_u} 1}.$$

Идея заключается в том, чтобы последовательно вычислять скрытые компоненты $H_{tr}(u)$ при заданном p_{tu} по формуле (7), а затем вычислять p_{tu} при заданном $H_{tr}(u)$ и так далее в итерациях, пока не будет достигнута сходимость (значения будут меняться мало).

Найдем начальное приближение для $H_{tr}(u)$, для этого выпишем еще раз выражение для скрытых компонент:

$$H_{tr}(u) = \frac{p_{tu} \frac{q_{tr} q_r}{\sum_{x \in T} q_{tx} q_x}}{\sum_{s \in T} p_{su} \frac{q_{sr} q_r}{\sum_{x \in T} q_{sx} q_x}}.$$

Пусть начальное приближение для p_{tu} — равномерное $p_{tu} = \frac{1}{|T|}$, тогда:

$H_{tr}(u) = \frac{q(r \mid t)}{\sum_{s \in T} q(r \mid s)}$, где $q(r \mid t) = \frac{q_{tr} q_r}{\sum_{r' \in R} q_{tr'} q_{r'}}$.

Показанным выше способом мы можем оптимизировать p_{tu} при фиксированном q_{tr}. Совершенно аналогичным способом можно решить задачу максимума правдоподобия (6) для формулы (5). Тогда мы сможем решить обратную задачу: оптимизировать q_{tr} при фиксированном p_{tu}.

Теперь можно выписать алгоритм.

3.2.3 Описание алгоритма

Вход:

|*U*| — число пользователей;

|*R*| — число ресурсов;

|*T*| — число тем в профиле;

$D = (u_i r_i)_{i=1}^{l}$ — выборка посещений (из нее получается матрица посещений $A_{|U| \times |R|}$;

Выход:

p_{tu} — профили пользователей;

q_{tr} — профили ресурсов;

Алгоритм:

1: построить списки D_u и D_r непустых значений в матрице *A*.

2: задать случайное начальное приближение для q_{tr} такое, что $\sum_{t=1}^{|T|} q_{tr} = 1, \forall r$.

3: вычислить $q_r = \frac{\sum 1}{|R|}$ и $p_u = \frac{\sum 1}{|U|}$ — средние значения для каждого пользователя и для каждого ресурса.

4: **повторять**

I. Оптимизируем p_{tu} при фиксированном q_{tr}:

5: **для всех** $t \in T$

6: вычислить суммы $S_t = \sum_{r' \in R} q_{tr'} q_{r'}$

7: **для всех** $r \in R$

8: вычислить $q(r \mid t) = \frac{q_{tr} q_r}{S_t}$

9: вычислить $H_{tr}(u) = \dfrac{q(r \mid t)}{\sum_{s \in T} q(r \mid s)}$ (не зависит от u)

10: **повторять**

11: **для всех** $u \in U$, $t \in T$

12: вычислить $p_{tu} = \dfrac{\sum_{r \in D_u} H_{tr}(u)}{\sum_{r \in D_u} 1}$

13: **для всех** $u \in U$, $r \in R$

14: вычислить суммы $S_{ur} = \sum_{s \in T} p_{su} q(r \mid s)$

15: **для всех** $t \in T$

16: вычислить $H_{tr}(u) = \dfrac{p_{tu} q(r \mid t)}{S_{ur}}$

17: **пока** не сойдется

II. Оптимизируем q_{tr} при фиксированном p_{tu}:

18: **для всех** $t \in T$

19: вычислить суммы $S_t = \sum_{u' \in U} p_{tu'} p_{u'}$

20: **для всех** $u \in U$

21: вычислить $p(u \mid t) = \dfrac{p_{tu} p_u}{S_t}$

22: вычислить $H^*_{tu}(r) = \dfrac{p(u \mid t)}{\sum_{s \in T} p(u \mid s)}$

23: **повторять**

24: **для всех** $r \in R$, $t \in T$

23: вычислить $q_{tr} = \dfrac{\sum_{u \in D_r} H^*_{tu}(r)}{\sum_{u \in D_r} 1}$

25: **для всех** $u \in U$, $r \in R$

26: вычислить суммы $S_{ru} = \sum_{s \in T} q_{sr} p(u \mid s)$

27: **для всех** $t \in T$

28: вычислить $H^*_{tu}(r) = \dfrac{q_{tr} p(u \mid t)}{S_{ru}}$

29: **пока** не сойдется

30: **пока** не сойдется

Данный алгоритм был опробован как на модельных, так и на реальных данных поисковой машины и дал неплохие результаты. Описание этих экспериментов будет дано ниже, а сейчас приведем описание идеи построения обобщенного профиля пользователей и ресурсов, как одного из возможных подходов к решению задач коллаборативной фильтрации.

3.3 Идея построения обобщенного профиля

Для всего множества ресурсов и пользователей определим ключевые свойства или темы, которые наиболее ярко выражают пользовательские интересы и свойства ресурсов. Вектор числовых значений, соответствующих каждой теме, назовем тематическим профилем ресурса (пользователя). В данном случае профили пользователей и ресурсов, вообще говоря, являются различными и могут содержать различное число тем.

В большинстве прикладных задач, мы можем автоматически оценить, хотя бы частично, вектор значений в тематическом профиле каждого вновь появившегося ресурса (пользователя). Это значит, что еще до того, как ресурс был выбран хоть одним пользователем (пользователь совершил какой-либо выбор), его профиль может быть оценен используя следующую информацию: описание данного ресурса при его регистрации в системе, тематический каталог ресурсов, частичную классификацию ресурсов, заполненную пользователем анкету со специально подобранными вопросами и т.д. Вся эта информация может быть использована для построения начального приближения для алгоритма, с помощью которого этот профиль будет уточняться по мере того, как по отношению к данному ресурсу будут выполняться какие-либо действия со стороны пользователей (или данный пользователь будет совершать какие-либо действия по выбору ресурсов). Профиль ресурса запишем как

$R_i = (\alpha_{i1}, \alpha_{i2}, ..., \alpha_{iN})$, где $\alpha_{i1}, \alpha_{i2}, ..., \alpha_{iN}$ — вес каждого признака для ресурса R_i.

Профиль пользователя:

$U_j = (\beta_{j1}, \beta_{j2}, ..., \beta_{jM})$, где $\beta_{j1}, \beta_{j2}, ..., \beta_{jM}$ — веса предпочтений (интересов) для пользователя *u*.

Будем считать, что для каждого нового ресурса или пользователя его профиль частично известен. Например, если признаком будет являться возраст, то он может быть изначально известен после заполнения пользователем регистрационной анкеты.

Для решения задач коллаборативной фильтрации нам необходимо каким-то образом оценивать близость ресурсов и пользователей. А так как в нашем случае они заданы своими профилями, стоит вопрос, а каким образом нам сравнивать эти профили? Ведь природа признаков ресурсов и предпочтений пользователей совершенно разная, хоть они и могут быть где-то схожими. Например, если ресурсами являются фильмы, то признаками для него могут являться жанр, режиссер, актерский состав и т.д. А для пользователя могут быть известны возраст, образование, социальный статус и т.д. Для того чтобы было возможно каким-то образом сравнивать профили пользователей и ресурсов предлагается ввести единый профиль, который будет объединять в себе и признаки ресурсов и предпочтения пользователей. Таким образом, в результате объединения фильм сможет получить признак «молодежный» или «дамский», а пользователь — «любитель боевиков».

Обобщенный профиль запишем следующим образом:

$P_i = (\alpha_{i1}, \alpha_{i2}, ..., \alpha_{iN}, \beta_{j1}, \beta_{j2}, ..., \beta_{jM})$

Нашей задачей будет дополнить для каждого пользователя и ресурса недостающую часть профиля. Тогда мы легко сможем решить поставленные выше задачи.

Действительно, если мы знаем для пользователя, какие признаки ресурсов для него наиболее близки, мы легко сможем проставить рейтинг любого ресурса, просто сравнив профили данного ресурса и данного пользователя, например, вычислив среднеквадратичное отклонение. Аналогично и для ресурса мы сможем найти потенциально близких пользователей. Построив некоторую функцию близости на профилях, мы автоматически получаем единую метрику на множестве пользователей и ресурсов, что удобнее, чем разделенные метрики.

С помощью такого подхода ресурсы, которые до этого не были выбраны пользователями ни разу, все равно могут быть включены в построение метрик, персональных предложений и т.д. Использование начальной информации поможет избежать проблемы «холодного старта», когда мы ничего не можем сказать о вновь появившемся ресурсе и никаким образом не можем его рейтинговать (советовать как потенциально интересный) для пользователя.

Данный подход здесь рассмотрен только как идея и является одним из перспективных направлений дальнейших исследований в рассматриваемой области.

4. Вычислительные эксперименты

4.1 Проверка алгоритма на модельных данных

Для проверки работы алгоритма была сгенерирована выборка посещений по заранее заданным профилям пользователей и ресурсов и средним значениям для пользователей и ресурсов.

Профили пользователей генерировались тремя способами: сначала случайной расстановок одной единички в каждый профиль (каждый пользователь интересуется только одной темой и каждый ресурс соответствует только одной теме), затем случайной расстановкой двум темам в каждом профиле значения 0.5, и, наконец, полностью случайным заполнением профилей.

Генерация данных происходила по следующему алгоритму:

1. Задаются априорные вероятности и профили для каждого пользователя и каждого ресурса
2. Вычисляются апостериорные распределение ресурсов $q(r|t)$ по формуле Байеса.
3. Вычисляются распределения ресурсов $p(r \mid u) = \sum_t p(t \mid u) q(r \mid t)$ для каждого пользователя u.
4. Затем начинается цикл генерации протокола посещений, на каждой итерации:
5. Согласно априорному распределению p_u случайно выбирается пользователь.
6. Согласно распределению ресурсов $p(r|u)$ выбирается ресурс.
7. Пара (u,r) заносится в протокол.

Для выработки критерия сходимости вычислялось среднее отклонение полученных профилей от исходных модельных. На графике показана зависимость среднеквадратичной ошибки от числа итераций во внешнем и внутреннем циклах. Каждая кривая соответствует определенному числу внешних итераций алгоритма, а по оси абсцисс отложены значения внутренних итераций.

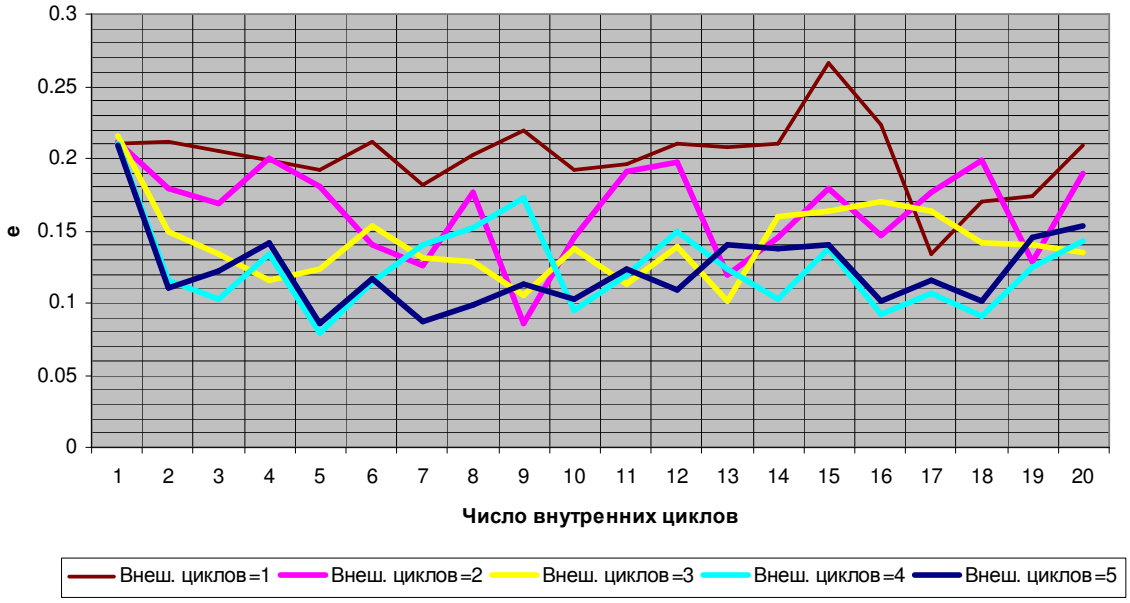


Рис.2 Среднеквадратичная ошибка от числа итераций во внешнем и внутреннем циклах. В исходных данных случайная расстановка единичек.

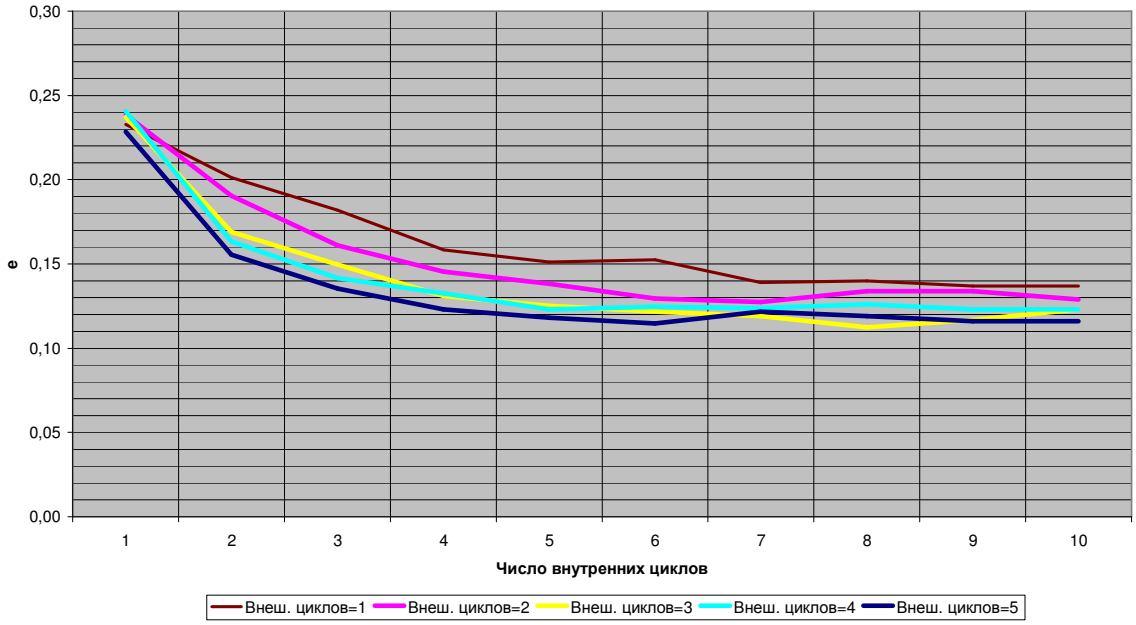


Рис.3 Среднеквадратичная ошибка от числа итераций во внешнем и внутреннем циклах. В исходных данных профили задаются случайным образом.

Из графика видно, что в случае случайного задания профилей, результирующие профили слабо меняют свои значения после 5-и внутренних итераций и 4-х внешних. Эти графики можно использовать для оценки критерия сходимости алгоритма.

Генерация данных происходила со следующими параметрами:

количество пользователей ***U=1000***;

количество ресурсов ***R=200***;

количество тем: ***T=10;***

количество сгенерированных событий типа «пользователь *u* посетил ресурс *r*» ***N=50000,***

q_r и p_u — распределены по гиперболическому закону.

4.2 Эксперимент на данных поисковой машины

Работа алгоритма была опробована на данных поисковой машины Yandex. Исходными данными являлись протоколы переходов пользователей со страниц результатов поиска. Протоколы охватывали 7 дней, по 5–10 миллионов запросов в день. Для каждого запроса в протоколе фиксировался уникальный идентификатор пользователя, список выданных документов и время обращения пользователя к выбранным документам. Исследуемый лог содержал данные о 14 606 пользователях и 129 600 посещениях.

Для анализа были отобраны около 1000 наиболее посещаемых ресурсов.

Для всех ресурсов были вычислены профили (с количеством тем ***T=20***). После чего была получена метрика на множестве ресурсов (вычислением среднеквадратичного отклонения на профилях). Для полученной метрики была применена монотонно возрастающая функция и отсечение больших значений по порогу так, чтобы расстояния равномерно распределились в промежутке [0,1].

Имея матрицу попарных расстояний, можно визуализировать множество ресурсов в виде карты сходства. Для этого применяется разреженный алгоритм многомерного шкалирования [7]. Он представляет множество объектов в виде точек на плоском графике таким образом, чтобы евклидовы расстояния между точками как можно точнее соответствовали заданным расстояниям.

Для построения Цветной Карты Сходства используется либо уже имеющийся каталог сайтов, либо выборка сайтов, классифицированная экспертом. Далее карта разбивается на области с помощью диаграммы Вороного [12]. На рисунке 4 хорошо видно, что ресурсы схожей темы, как правило, образуют группы точек, находящиеся внутри областей соответствующего цвета. Каждая такая группа точек соответствует ресурсам схожей темы, несмотря на то, что информация от контенте ресурсов при построении карты вообще не использовалась.

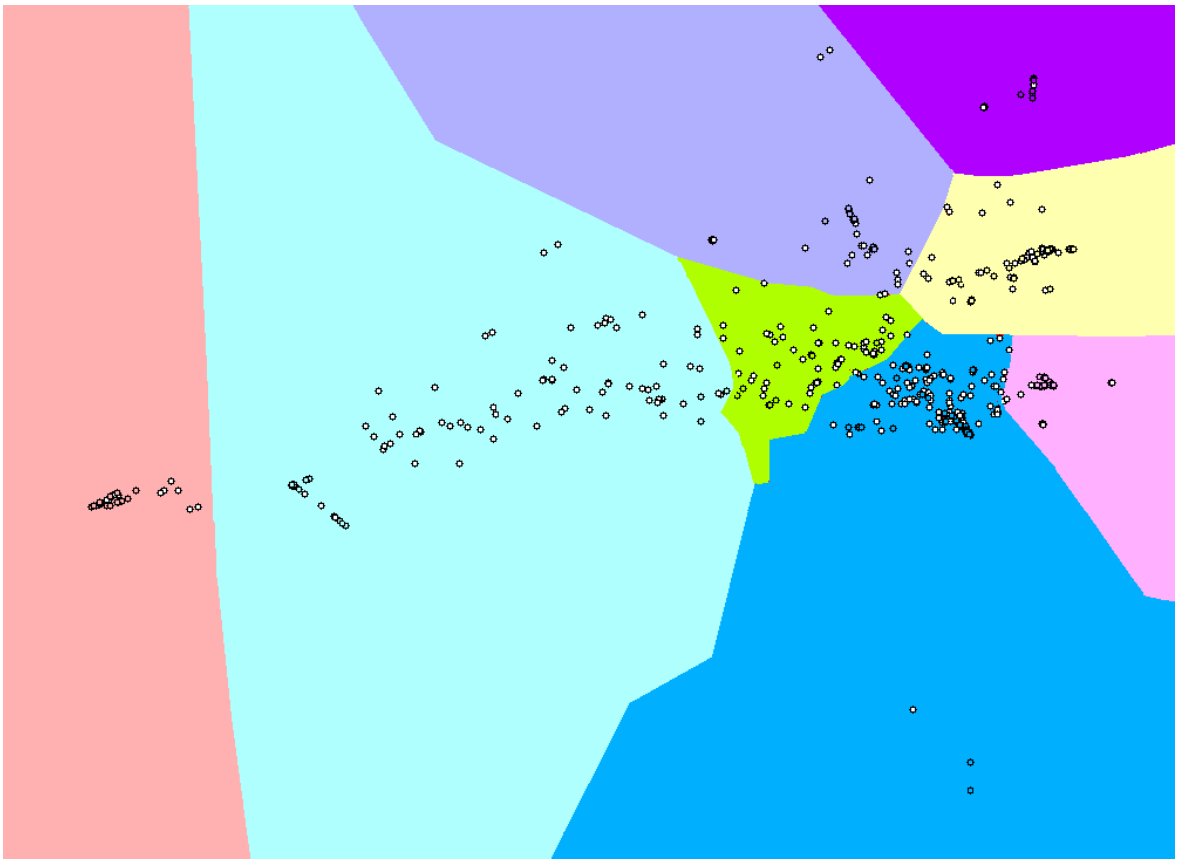


Рис.4 Карта сходства ресурсов, вычисленная по профилям ресурсов

В качестве наиболее объективного критерия качества построенной метрики применяется метод k ближайших соседей. Качество построения оценивается по количеству ошибок при попытке классифицировать точки методом kNN, используя частичную классификацию ресурсов.

Сначала производится настройка параметра алгоритма kNN (рис. 5). Сиреневая кривая на графике отражает качество метрики, построенной точному тесту Фишера. Красный график отражает ошибку описанного выше метода, основанного на EM-алгоритме. Как видно, качество построения метрики алгоритмом, основанным на профилях, значительно лучше. Из графика также видно, что оптимальное значение функционала возникает при количестве рассматриваемых соседей $k = 12$.

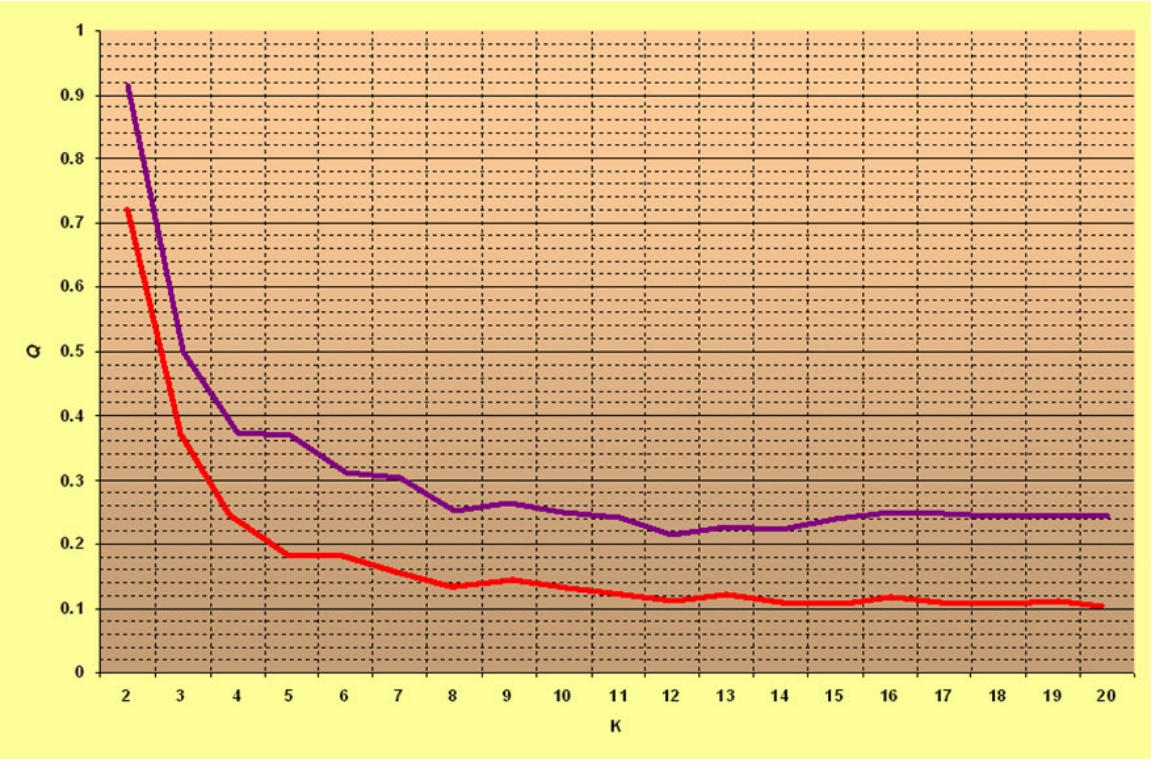


Рис.5 Оптимизация количества соседей в методе kNN и сравнение двух алгоритмов: точного теста Фишера (сиреневая кривая) и сравнение профилей ресурсов (красная кривая).

Для точного теста Фишера также возможно произвести настройку параметра алгоритма построения метрики (рис. 6). Для различных значений порога информативности считается функционал качества и как видно на графике (синяя кривая) при некотором значении информативности достигается минимум функционала Q.

Для проверки качества построения карты сходства и оценки искажений, вносимых в исходную метрику, аналогичный функционал строится для евклидовых расстояний на плоскости (красная кривая). Как видно из графика, построение карты сходства методом многомерного шкалирования заметно занижает качество исходной метрики, однако дает возможность наблюдать основные закономерности метрики. Как видно из графика, оптимальная карта сходства получается при пороге информативности равном 0.15.

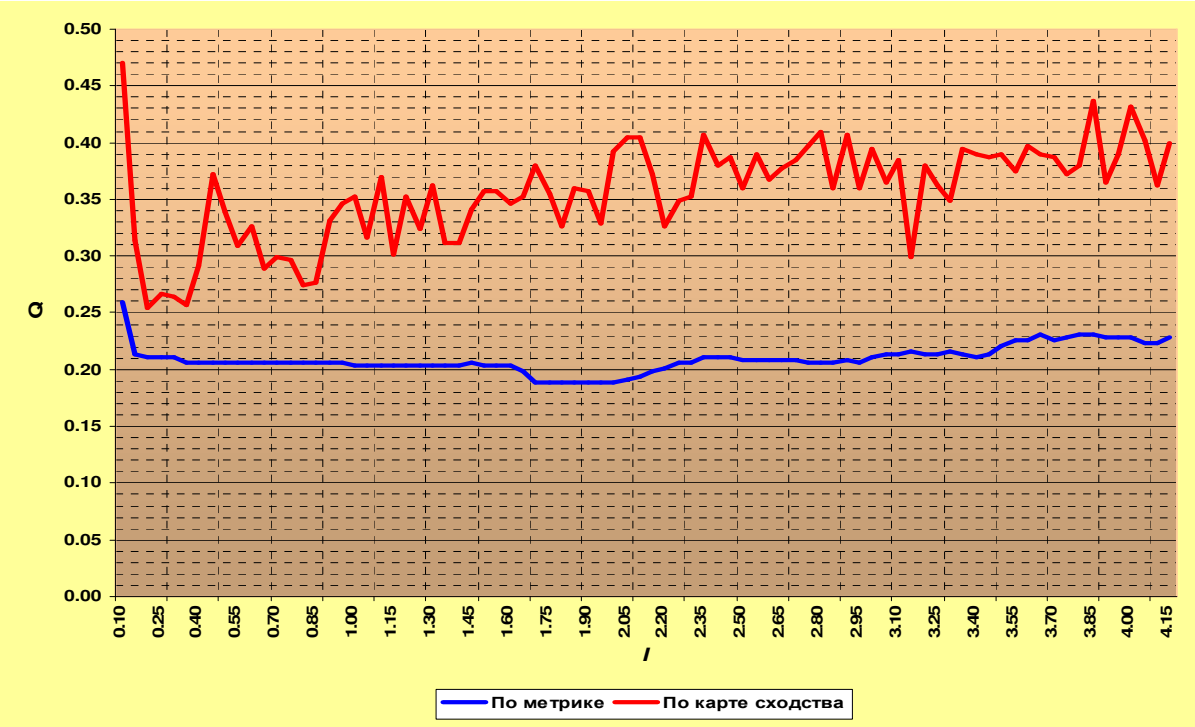


Рис.6 Оптимизация порога информативности методом kNN при k=12

Для сравнения приведем пример карты сходства, полученной алгоритмом, в основе которого лежит предположение, что посещения ресурсов r и r' являются независимыми событиями и количество чисто случайных посещений обоих ресурсов одним и тем же пользователем подчиняться гипергеометрическому распределению (точный тест Фишера).

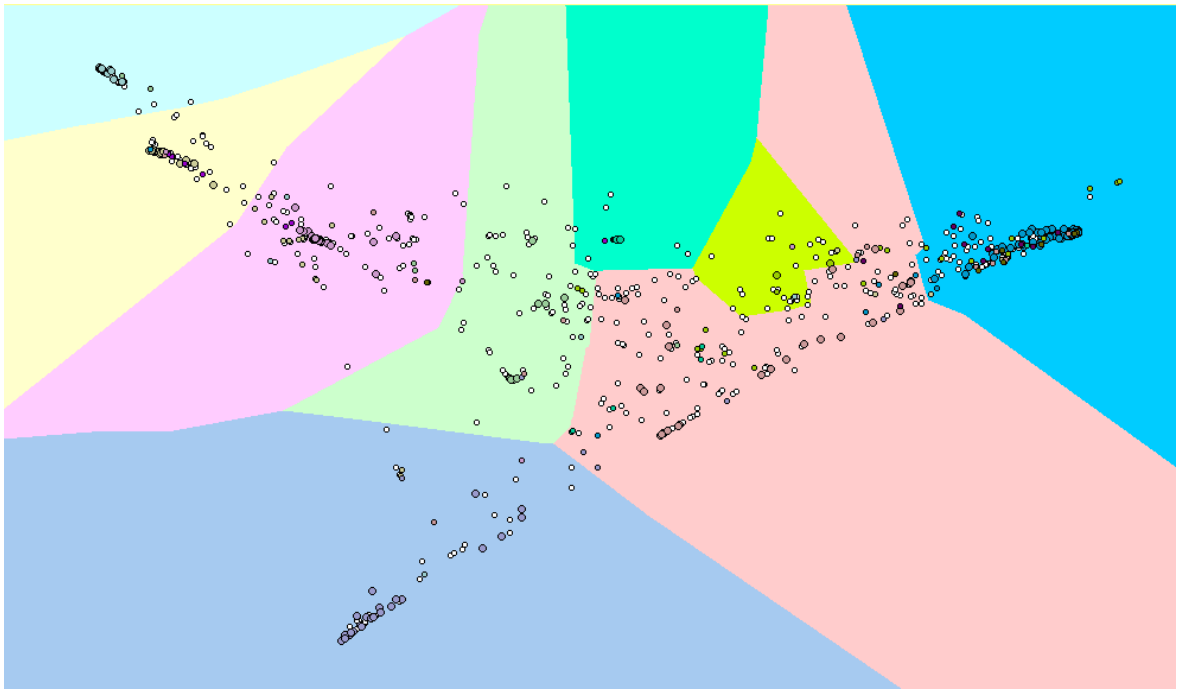


Рис. 7. Карта сходства ресурсов для точного теста Фишера

5. Заключение

В результате проделанной работы удалось выяснить, что

1) Предложенный алгоритм, основанный на восстановлении профилей пользователей и ресурсов, позволяет значительно сократить объем хранимых в памяти данных, ускорить обработку данных, повысить качество получаемых метрик.
2) Результатами алгоритма являются профили, поддающиеся содержательной интерпретации.
3) Алгоритм легко приспособить для учета априорной информации о ресурсах или пользователях путем формирования соответствующего начального приближения для профилей.
4) Решается проблема «холодного старта», когда мы ничего не можем сказать о вновь появившемся пользователе и ресурсе. Для нового ресурса или пользователя профиль может быть задан из априорных соображений.
5) Данный алгоритм имеет более широкий спектр применений. Например, появляется возможность сравнивать не только ресурсы с ресурсами и пользователей с пользователями, но и пользователей с ресурсами.

Описанная методика может быть развита и применена для ряда важных задач, исследование которых не вошло в рамки данной работы. Интересными направлениями дальнейших исследований являются:

1) Построение иерархических профилей.
2) Оптимизация тематической структуры профиля.
3) Построение обобщенных профилей, в которых объединяются тематики и социально-демографические характеристики пользователей.

Список литературы

[1] Review of Personalization Technologies: Collaborative Filtering vs. ChoiceStream's Attributized Bayesian Choice Modeling. Technology Brief.

[2] Лекции по статистическим (байесовским) алгоритмам классификации. К. В. Воронцов

[3] Forecsys CEA::WUM. К. В. Воронцов

[4] The Expectation Maximization Algorithm. Frank Dellaert. College of Computing, Georgia Institute of Technology. Technical Report number GIT-GVU-02-20. February 2002

[5] A Scalable Collaborative Filtering Framework based on Co-clustering. Thomas George, Department of Computer Science, Texas A & M University; Srujana Merugu, Department of Electrical and Computer Engineering, University of Texas at Austin

[6] Collaborative Filtering with the Simple Bayesian Classifier. Koji Miyahara and Michael J. Pazzani

[7] Лексин В.А. Методы выявления взаимосогласованных структур сходства в системах взаимодействующих объектов, ВКР Бакалавра.

[8] Технология анализа клиентских сред. http://www.forecsys.ru/cea.php. Форексис. 2005.

[9] Воронцов К. В., Вальков А. С. О быстрых алгоритмах синтеза плоских представлений метрических конфигураций. Искусственный Интеллект, Донецк, 2004. №2 с.43–48.

[10] Выявление и визуализация метрических структур на множествах пользователей и ресурсов Интернет. В. А. Лексин, К. В. Воронцов (тезисы для конференции ИОИ-2006).

[11] О метрических структурах на множествах пользователей и ресурсов Интернет. ВоронцовК.В., РудаковК.В., Лексин В.А., Ефимов А.Н (тезисы для конференции ММРО-12)

[12] Скелетизация многосвязной многоугольной фигуры на основе дерева смежности ее границы. Л.М. Местецкий