# Оглавление

[Оглавление 3](#_Toc295700917)

[Введение 5](#_Toc295700918)

[Глава 1. Рекомендательные системы 9](#_Toc295700919)

[1.1. Ухудшающийся отбор в культуре 9](#_Toc295700920)

[1.2. Цели и задачи рекомендательных систем 10](#_Toc295700921)

[1.3. Виды рекомендательных систем 11](#_Toc295700922)

[1.3.1. Контентные методики 13](#_Toc295700923)

[1.3.2. Методики совместной фильтрации 16](#_Toc295700924)

[1.3.3. Гибридные методики 20](#_Toc295700925)

[1.3.4. Метод скрытых факторов 22](#_Toc295700926)

[1.4 Выводы 24](#_Toc295700927)

[Глава 2. Метод вложенных тегов 26](#_Toc295700928)

[2.1. Представление данных 26](#_Toc295700929)

[2.2. Сравнение объектов 29](#_Toc295700930)

[2.3. Поиск объектов 35](#_Toc295700931)

[Глава 3. Экспериментальный стенд 38](#_Toc295700932)

[3.1. Подсчет среднеквадратичной ошибки функции 38](#_Toc295700933)

[3.1.1. Используемые данные 38](#_Toc295700934)

[3.1.2. Функция сравнения 42](#_Toc295700935)

[3.1.3. Предсказание оценки 42](#_Toc295700936)

[3.1.1. Вывод 42](#_Toc295700937)

[3.2. Обосновывающая рекомендательная система 43](#_Toc295700938)

[3.1.1. Используемые данные 43](#_Toc295700939)

[3.1.2. Функция сравнения 43](#_Toc295700940)

[3.1.3. Архитектура системы 44](#_Toc295700941)

[3.1.4. Внешний вид системы и элементы управления 45](#_Toc295700942)

[3.1.1. Вывод 46](#_Toc295700943)

[Заключение 47](#_Toc295700944)

[Источники 48](#_Toc295700945)

# Введение

*«Мы достигли точки критической массы, когда количество памяти, которую мы экстериоризировали в книги и базы данных (перечислил несколько источников), сейчас превышает количество памяти, хранящейся внутри совокупности наших биологических тел. Другими словами, «снаружи» памяти больше, чем существует внутри «всех нас». Мы перепрограммировали свою сущность».*

Дуглас Коупленд, «Рабы Майкрософта».

*«Большинство людей на вопрос «Ваш любимый фрукт, поэт и цветок» отвечают «яблоко, Пушкин и роза». Можем ли мы, основываясь на этой информации, рекомендовать читать Пушкина всем любителям яблок?».*

**Актуальность проблемы.** За все время своего существования, интернет и смежные с ним сообщества сагрегировали в себе немыслимые объемы информации. Для того чтобы проще ориентироваться в потоке этой информации разработчики придумывают разнообразные средства поиска и фильтрации информации.

Огромным шагом в организации глобального информационного пространства стало создание поисковых систем. Первой такой системой стал Wandex в 1993 году. После него появился Aliweb, работающий и по сей день. Прошло около пятнадцати лет и поисковые системы настолько плотно вошли в жизнь, что в речи стали появляться глаголы, образованные от названий поисковых систем, их стали упоминать в литературе и кино.

Поисковые системы позволяют найти страницы (а в последнее время узкоспециализированную информацию: музыку, картинки, товары, магазины, научные статьи) по ключевым словам. Они помогают пользователю найти то, о существовании чего он уже знает. Ситуация сильно осложняется, когда пользователь не можете заранее определить ключевые слова для поиска, потому что не обладает конкретной информацией о том, что хочет найти.

К 90-м годам интернет достаточно наполнился всевозможным содержимым, чтобы пользователи озадачились новым вопросом – «я хочу найти то, что мне интересно, но не знаю что именно», на который поисковая система не способна ответить. Как раз в этот момент начинают развиваться рекомендательные системы. Они берут за основу имеющуюся у них информацию о пользователе и предлагают ему то, что возможно его заинтересует.

С момента появления первых работ по коллаборативной фильтрации в середине 1990-х годов, рекомендательные системы стали объектом пристального научного внимания. В течение последнего десятилетия была проделана большая работа как теоретического, так и прикладного характера, посвященная развитию рекомендательных систем. В настоящее время проблема рекомендательных систем сохраняет к себе большой интерес, так как в этой области остается много задач, решение которых сулит множество возможностей практического применения, что должно помочь пользователям справляться с громадным объемом информации, а также снабдить их инструментами выработки персонифицированных рекомендаций. Примером конкретного применения рекомендательных систем может служить система рекомендаций книг, CD и других товаров на Amazon.com [3], фильмов на MovieLens [4], новостей на VERSIFI Technologies (бывшем AdaptiveInfo.com) [5].

Современные рекомендательные системы позволяют упорядочить неизвестные вам объекты в системе таким образом, чтобы наиболее интересные оказались в начале списка. Например, если объектами будут являться блюда в ресторане, то рекомендательная система отсортирует все меню соответствующим образом, но смешает блюда из всех разделов в одном списке. Как, руководствуясь этим списком, выбрать себе еду? Предположим, что в основном вы предпочитаете мясные блюда, но сегодня хотели бы что-нибудь из рыбы. Как понять, какие из интересующих вас объектов интересны вам в данный момент и почему? Как построить рекомендательную систему, которая способно была бы фильтровать список рекомендуемых объектов и предоставить вам только рыбные блюда?

**Цель работы** — предложить метод построения обосновывающей рекомендательной системы с функциональностью фильтрации рекомендаций.

**Основные задачи.** Для достижения указанной цели работы решены следующие задачи:

1. Провести обзор существующих методов построения рекомендательных систем, выявить возможности по созданию обосновывающих рекомендательных систем.
2. Предложить метод построения обосновывающей рекомендательной системы с возможностью фильтрации рекомендаций.
3. Построить экспериментальный стенд выяснить, возможно ли применение предложенного метода в промышленных системах.

**Научная новизна.** Научная новизна предлагаемого метода состоит в возможности построения обосновывающей рекомендательной системы с возможностью фильтрации рекомендаций при меньших затратах ресурсов.

**Практическое значение** работы заключается в том, что разработанный метод примени для построения промышленных рекомендательных систем.

**Апробация** была проведена с использованием данных исследовательской группы MovieLens и данных базы данных о фильмах IMDB.

**Структура работы.** Работа изложена на 52 страницах, состоит из введения, трех глав и заключения. В главе 1 представлен обзор предметной области: существующих методов для построения рекомендательных систем, перечислен список их недостатков, сделаны выводы о том, почему они не позволяют строить обосновывающие рекомендательные системы. В главе 2 описывается разработанный метод для построения обосновывающей рекомендательной системы с возможностью фильтрации рекомендаций. В главе 3 описывается построение экспериментального стенда и проведение эксперимента с целью выяснения возможности применения разработанного метода в промышленных системах. В заключении подводятся итоги работы и выявляются потенциальные направления дальнейших исследований.

# Глава 1. Рекомендательные системы

В данной главе рассматриваются виды рекомендательных систем, их цели и задачи.

## 1.1. Ухудшающийся отбор в культуре

В последнее время все чаще наблюдаются процессы «ухудшающего отбора в культуре». Возникновение таких процессов описал Джордж Акерлоф в 1970 году [1]. За объяснение этих процессов он получил Нобелевскую премию в 2001 году.

Суть их заключается в том, что качественная продукция постепенно исчезает с рынка, в то же время некачественная - становится нормой. Это происходит вследствие того, что потребитель обладает меньшей информацией о продукте по сравнению с продавцом. Акерлоф рассматривал вторичный рынок автомобильной промышленности, он продемонстрировал, что происходит, если на рынке отсутствует честное экспертное мнение. Продавать старые, отработавшие свое машины, под видом новых куда более выгодно, чем новые, а неопытный потребитель не в состоянии отличить приличный автомобиль от плохого и не готов платить справедливую цену за первый. В итоге рынок разрушается, на нем остаются только плохие машины. По той же схеме развивается принцип ухудшающего отбора на рынках культурных продуктов. Из экономических соображений качественный продукт требует больших затрат, поэтому производители предпочитают незначительные увеличения бюджета маркетинговых акций значительным увеличениям затрат на производство. Таким образом, покупатель оказывается в ситуации, где он обязан заплатить за товар, не зная, насколько он ему понравится, а продавец способен его обмануть, используя различные маркетинговые технологии.

## 1.2. Цели и задачи рекомендательных систем

Основная цель рекомендательных систем – решить проблему ухудшающего отбора в культуре. Иными словами, предоставить пользователю независимую экспертную оценку о продукте, который заинтересует его, или составить предположительное мнение пользователя обо всех имеющихся продуктах. По целям рекомендательные системы можно разбить на несколько классов (на основе статьи [2]).

1. *Аннотация в контексте.* Первоначальным сценарием рекомендательных систем была фильтрация всех сообщений на некотором форуме с целью принятия решения, какое из них читать. Эта задача требовала сохранения порядка и контекста сообщений, и соответствующим образом использовала предсказания для аннотирования сообщений в их контексте. В некоторых случаях «самые плохие» сообщения отфильтровывались.
2. *Найти хорошие объекты.* Это, пожалуй, является задачей большинства существующих рекомендательных систем. Требуется предоставить пользователю ранжированный список объектов с предполагаемыми оценками пользователя. Во многих коммерческих системах показываются только наиболее выигрышные объекты, а прогнозируемые значения оценок нет.
3. *Найти все хорошие объекты.* Такая задача может быть актуальна в рекомендательной системе по безе судебных дел. Для юриста, ищущего прецеденты, скорее важнее не пропустить ни одного случая, чем ранжировать их в нужном порядке. В первую очередь такие рекомендательные системы должны гарантировать низкую вероятность ошибочных заключений (минимизировать число заключений, нужных пользователю, но кажущихся бесполезными системе).
4. *Последовательность рекомендаций.* С растущим объемом музыки, доступной в интернете, растет актуальность этой проблемы. «Здесь возникает проблема перехода от рекомендаций одной песни к рекомендации целой последовательности музыкальных композиций, которую будет приятно прослушать». [2]
5. *Только просматривание.* Обычно рекомендательная система оценивается на основе того, как хорошо она подбирает объекты для пользователя. «В беседах с пользователями MovieLens, Amazon и некоторых других сайтов авторы статьи обнаружили, что многие из них пользуются РС даже тогда, когда у них нет намерения приобрести вещь» [2]. Некоторым пользователям приятно просматривать информацию о рекомендуемых объектах, а для других – это процесс познания.

## 1.3. Виды рекомендательных систем

В самом общем виде проблема выработки рекомендаций сводится к присвоению той или иной оценки товару, еще не известного потенциальному покупателю [6]. Очевидно, такая оценка дается исходя из анализа предшествующих предпочтений данного покупателя или любой другой информации о нем, формальное описание которой дано ниже. После того, как система предсказывает оценки для еще не известных потребителю товаров, те из них, которые получают наивысшие оценки рекомендуются потенциальному потребителю.

Формально проблема выработки рекомендаций может быть представлена следующим образом: пусть — группа ( множество) всех пользователей, — группа (множество) предлагаемых товаров (книг, фильмов, ресторанов и т.д.). Группа всех возможных товаров может быть очень объемной, достигая в некоторых областях сотен тысяч и даже миллионов единиц товаров (книги, CD); группа пользователей также может достигать чрезвычайно больших цифр. Пусть – функция полезности, описывающая полезность предмета для , т.е. , где – количество заказанных единиц товара (то есть любое положительное целое число). Тогда для каждого потребителя мы хотим выбрать такой товар , который наиболее соответствовал бы полезности потребителя. Более формально:

В зависимости от контекста, полезность может быть определена потребителем, как это обычно бывает в системах, основывающихся на потребительской оценке (в частности, количеством баллов, отражающих, насколько конкретному покупателю понравился конкретный товар, например, Иван Пупкинс дал фильму Гарри Потттер 7 баллов из 10), или может быть вычислена программно, как в случае с функцией полезности, основывающейся на прибыли.

Каждый участник из группы потребителей может быть описан с помощью индивидуального профиля, содержащего различные характеристики потребителя: возраст, пол, доход, семейное положение и т.д. В самом простом случае, профиль может содержать единичную характеристику – ID потребителя. Или, например, в системе рекомендации фильмов, где это совокупность фильмов, каждый фильм может быть представлен не только своим ID, но и названием, жанром, режиссером, годом, главными актерами.

Новые оценки еще не оцененных товаров могут быть сделаны различными способами, такими как машинное самообучение, эвристическим методом, теорией аппроксимации. РС различаются в подходах к анализу оценок. Современная классификация делит РС на следующие категории в зависимости от того, как делаются рекомендации:

* *Контентные рекомендации*: потребитель получит рекомендации товаров, сходных с теми, которые он выбрал ранее.
* *Коллаборативные рекомендации*: потребителю будут предложены товары, в прошлом выбранные людьми со схожими с ним вкусами и предпочтениями.
* *Гибридные рекомендации*, сочетающие в себе два предыдущих метода.
* *Факторный анализ* позволяет предсказать оценку фильма на основе сингулярного разложения матрицы оценок. Предполагая наличие в системе определенного количества факторов можно получить степень выраженности этих факторов для каждого пользователя и объекта. Те объекты, в которых факторы проявляются близко к тому, как они проявляются у пользователя, рекомендуются ему.

### 1.3.1. Контентные методики

Данный подход основан на измерении похожести объектов между собой. Иными словами, вывод о полезности товара s для потребителя делается, исходя из полезности , присвоенной потребителем c товарам сходным с товаром . Схематичное изображение этого принципа представлено на рис. 1, где зеленым кругом обозначен потребитель, красными товары с присвоенной функцией полезности, а белым кругом с красным — контуром товар, для которого строится предположительная оценка.

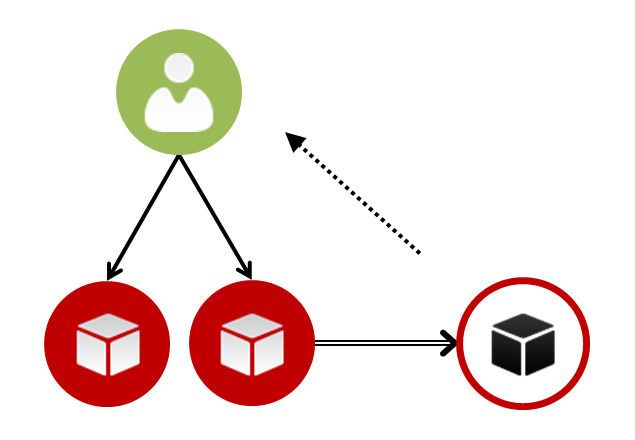


Рис. . Принцип контентных методик

Например, в системе, рекомендующей фильмы, для того, чтобы рекомендовать фильмы потребителю , контентная РС пытается найти сходство между фильмами, высоко оцененными потребителем ранее (общие актеры, режиссеры, жанры и т.д.) И только фильмы, обладающие высокой степенью общности с предпочтениями потребителя, будут рекомендованы. Контентный подход рекомендательных систем основан на работах, изучающих методы поиска информации [8, 9] и информационной фильтрации [10]. Так как наибольшие успехи были достигнуты в области текстового поиска и фильтрации, многие современные рекомендательные системы работают с объектами, содержащими текстовую информацию, такие как документы, веб страницы, сообщения сети Usenet.

Например, такой подход был реализован а рекомендательной системе PRES (сокращение от Personal Recommender System, [7]). Система PRES специализируется на рекомендациях интересных (релевантных для выбранного пользователя) страниц внутри одного сайта. Разделяются понятия навигационных страниц и страниц-данных. (Страницы так же могут иметь смешанный тип). PRES выбирает объекты, исходя из корреляции между его содержимым и профилем пользователя, в котором собраны его предпочтения, что полностью противоположно принципу коллаборативной фильтрации, который основывается на измерении похожести пользователей между собой. В ходе работы с системой измеряется заинтересованность пользователя в том или ином документе, которые учитываются в его профиле. Далее система подбирает наиболее релевантные документы из имеющейся коллекции, основываясь на критериях, указанных в пользовательском профиле.

Контентные методики обладают рядом проблем:

* *Ограниченность контентного анализа*. Для адекватной работы системы контент должен либо обладать формой, доступной для автоматического машинного анализа, либо должен назначаться вручную [6]. Системы поиска информации вполне успешно находят заданные характеристики в тексте, в то время как анализ иных свойств объекта сталкивается с определенными сложностями. Например, куда более трудоемким оказывается анализ мультимедийных объектов (музыки и фильмов). Подобные ограничения делают малоэффективным назначения свойств вручную.
* *Проблема узких рекомендаций*. Если система рекомендует только те товары, чьи характеристики совпадают с содержащимися в профиле потребителя, это означает, что он получит рекомендации только таких товаров, которые сходны с товарами, ранее уже получившими его оценку [6]. Например, человек, никогда не имевший дела с греческой кухней, никогда не получит совета посетить греческий ресторан.
* *Проблема неоднородности контента*. В случае, когда система покрывает несколько направлений контента (скажем, рекомендательная система для музыки, фильмов и ресторанов), то оценки, полученные для одного вида контента не могут быть использованы для расчёта рекомендаций для другого вида контента.

Ограниченность анализа в совокупности с проблемой узких рекомендаций не позволяют строить обосновывающие рекомендательные системы. Функции, используемые для сравнения объектов (например, используемые ключевые слова в статье или гармонии аудиопотока) не позволяют наглядно обосновать причину рекомендаций объекта, а значит, не позволят использовать в качестве критерия фильтрации.

### 1.3.2. Методики совместной фильтрации

В создании рекомендательных систем этот подход более популярен, чем подход с применением контентно-ориентированной фильтрации. Основная идея данного подхода заключается в составлении рекомендаций оценок для данного пользователя на основе мнения остальных. Оценки других пользователей могут быть получены из их профилей явно или неявно.

Более формально, товара s для пользователя c оценивается исходя из полезностей приписанных товару s теми пользователями , которые «похожи» на пользователя . Этот принцип схематично изображен на рис. 2, где зеленым кругом обозначен пользователь, а белым кругом с зеленым контуром «похожий» на него пользователь.

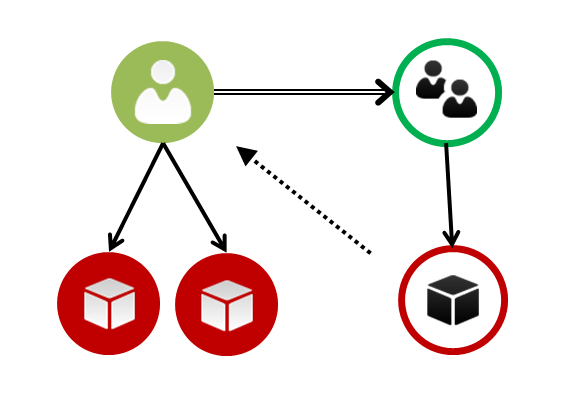


Рис. . Принцип работы контентных методик

Множество коллаборативных систем было разработано в бизнесе и в академической науке. Считается, что первой была система Grundy, предложившая использовать стереотипы поведения для построения моделей клиентов, основываясь на небольшом количестве информации о каждом клиенте. Используя стереотипы поведения клиентов, Grundy создавала клиентские профили и использовала их для рекомендации подходящих книг каждому клиенту. Позже система Tapestry использовала индивидуальный анализ для ручного поиска клиентов, обладающих похожими вкусами. GroupLens, Video Recommender, and Ringo были первыми системами, использовавшими алгоритмы коллаборативной фильтрации для автоматического предсказывания. Среди других рекомендательных систем, использующих коллаборативную фильтрацию, можно назвать систему, рекомендующую книги на Amazon.com, систему PHOAKS, помогающую находить нужную информацию на WWW, систему Jester (шутник), советующую анекдоты.

В [15] алгоритмы, построенные по принципу коллаоративной фильтрации можно разделить на два основных класса: анамнестические (memory-based) и модельные (model-based).

Анамнестические алгоритмы [15, 16, 17], делают предсказания об оценках пользователя, исходя из его предшествующих оценок. Модельные же алгоритмы [18, 15, 19, 20, 21] по совокупности оценок изучают модель, на основании которой будут сделаны предсказания о будущих оценках. Кроме того, в [16] предлагается метод коллаборативной фильтрации, основанный на методах машинного самообучения, где работают разные системы машинного самообучения (например, система нейронных сетей), объединенные с методами извлечения релевантных признаков (таких как алгебраические модели сокращения матриц до матриц меньшего размера с сохранением репрезентативности).

Коллаборативные рекомендательные системы в чистом виде лишены недостатков контентных систем. В частности, поскольку коллаборативные системы используют рекомендации (оценки), сделанные другими пользователями, они могут работать с самым разным контентом и рекомендовать самые различные товары, включая товары, не имеющие сходства с приобретенными (или оцененными ранее).

Тем не менее, коллаборативные методики обладают рядом проблем:

* *Проблема нового пользователя.* Проблема сходна с проблемой контентных систем. Чтобы делать точные рекомендации, система должна прежде всего изучить предпочтения пользователя на основании данных им оценок [6].
* *Проблема нового товара.* Новые товары регулярно добавляются в рекомендационные системы. Коллаборативные системы при выработке рекомендаций руководствуются только предпочтениями пользователей. Поэтому рекомендационная система не может рекомендовать товар, пока он не получит достаточное количество оценок [6].
* *Разреженность.* В любой рекомендательной системе, количество оценок, которые необходимо предсказать, обычно намного превышает количество данных оценок. Важно, чтобы система умела эффективно предвидеть оценки, исходя из небольшого количества примеров [6].
* *Проблема изменчивости профиля.* В соответствии с исследованиями на протяжении последних лет, было показано, что коллаборативные рекомендательные системы хорошо работают для пользователей с однородными предпочтениями (вкусами). Если в достаточно однородную систему ввести пользователя с разнообразными предпочтениями, его оценки будут влиять на рекомендации для остальных пользователей, предлагая им неожиданные, и - чаще всего - неинтересные товары. Таким образом, с одной стороны, решена проблема узких рекомендаций, с другой стороны, есть риск перегрузить пользователя слишком разнообразными и неинтересными ему рекомендациями, что может привести с его стороны к отказу от пользования системой в целом. Так же, пользователь может проявлять неоднородность в предпочтениях к самому себе с течением времени, когда меняются его предпочтения, вкусы и, соответственно, профиль.
* *Масштабируемость.* Одна из самых главных проблем чистого коллаборативного подхода заключается в его ресурсоемкости (требование к памяти) и сложности (количество операций, необходимых для пересчета рекомендаций для пользователя), что делает затруднительным использование коллаборативных методик в системах с большим количеством пользователей.
* *Проблема первой оценки.* Система оценивает степень сходства товаров на основе проставленных для них оценок, поэтому, когда появляется новый неизвестный товар, система никогда его не порекомендует, пока кто-нибудь его не оценит. И это ведет нас к следующей проблеме.
* *Проблема избирательности внимания.* В случае нестатичной системы, при появлении новых товаров, наблюдается тенденция накопления большого количества оценок для уже существующих товаров, и отсутствия оценок для новых/малоизвестных товаров. Это ведет к тому, что известные существующие товары становятся все более и более популярными (чаще рекомендуются, чаще получают оценку пользователей, и как следствие, опять чаще рекомендуются), а малоизвестные новые товары никогда не предлагаются пользователям.

Коллаборативные методики основываются на понятии «похожих» пользователей, которые алгоритмически ищутся среди всех пользователей, присутствующих в системе. Чаще всего найденные «похожие» пользователи оказываются незнакомыми тому пользователю, для которого ищутся рекомендуемые объекты и не могут быть использованы в качестве явного обоснования рекомендаций. Так же пользователю затруднительно использовать других незнакомых ему пользователей для формулировки критериев фильтрации рекомендаций.

### 1.3.3. Гибридные методики

Гибридные рекомендательные системы появились как попытка решить проблемы обеих контентных и коллаборативных методик [12]. По способу объединения описанных выше подходов гибридные методы можно условно разделить на три класса:

1. Реализация по отдельности коллаборативных и контентных алгоритмов и объединение их предсказаний.
2. Добавление контентных свойств к коллаборативной модели.
3. Добавление коллаборативных свойств к контентной модели.

Первый способ чаще всего комбинирует результаты двух систем, прибегнув к линейной комбинации оценок или к схеме голосования.

Алгоритмы второго рода, например Fab [13], основываются на обычной коллаборативной фильтрации, но так же используют контентные профили пользователей. Это позволяет избежать проблемы разреженности оценок, свойственных коллаборативным алгоритмам, поскольку в действительности не так много пользователей номинируют одни и те же товары. Некоторые [14] обращаются к аналогичному подходу, используя различные т.н. filterbots - своего рода автоматические роботы, присваивающие определенную оценку только что появившимся документам или товарам.

В третьей категории наиболее распространенным подходом является методика сокращения выборки применительно к базе контентных профилей. Например, использование латентно-семантической индексации сделает возможным коллаборативный анализ совокупности клиентских профилей, если эти профили представлены в векторном виде, что может сделать контентный метод более эффективным.

Основными плюсами гибридных систем является то, что были решены проблемы:

* нового пользователя
* нового товара
* значительно решена проблема масштабируемости

Однако, сохраняются проблемы:

* разреженности
* первой оценки
* избирательности внимания
* изменчивости профиля
* неоднородности контента

### 1.3.4. Метод скрытых факторов

В 2006 компания Netflix объявила соревнование на лучший алгоритм предсказания оценки, которую зритель поставит фильму, на основе предыдущих оценок этого и других зрителей. Главный приз составлял $1000000, а для его получения надо было улучшить их собственный алгоритм Cinematch на 10%. Приз выдан команде BellKor’s Pragmatic Chaos 21 сентября 2009 года.

Команды, занявшие первые два места использовали алгоритмы, построенные с использованием скрытых факторов. Для подсчета функции используется факторный анализ, общий принцип работы метода представлен на рис. 3.

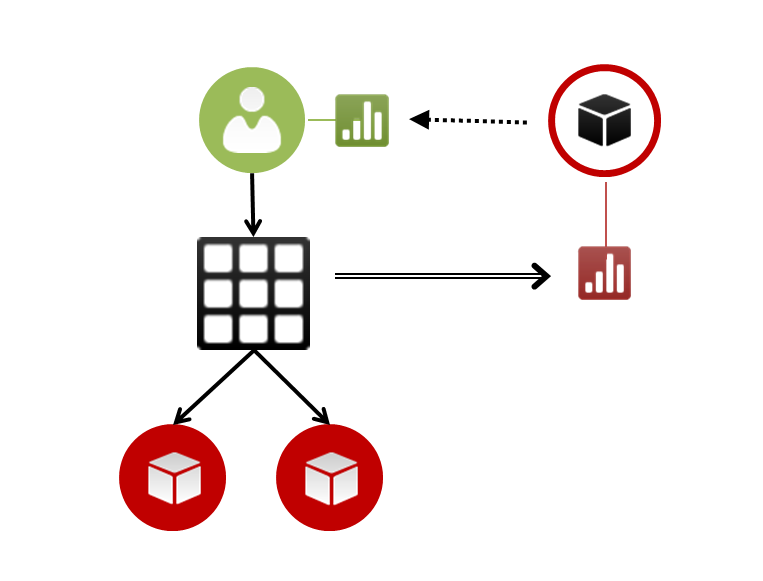


Рис. . Принцип работы метода скрытых факторов

Имея, например, два фактора, объекты и пользователей можно представить в виде векторов или точек на двумерной плоскости, где каждая координата будет соответствовать степени проявления каждого фактора. Так, факторов образуют пространство размерности , а каждому объекту будет соответствовать вектор , а пользователю c вектор , где вектора содержать информацию о релевантности каждого фактора для объекта и пользователя соответственно. Это позволяет легко получить оценку полезности объекта для пользователя следующим образом:

Таким образом можно получить оценку функции для любой пар пользователь-объект. Для получения степени выраженности может использоваться сингулярное разложение матрицы, но оно слишком ресурсоемко для работы с матрицей оценок, которая сильно разрежена, т.к. пользователи обычно отмечают лишь малую часть имеющихся объектов. Для этого применяются другие эвристически методы достижения данной цели, которые подробно описаны в статье [32].

Такой метод позволяет получить более точные оценки, чем контентные методы и методы коллаборативной фильтрации, но, тем не менее, обладает следующими недостатками:

* Даже при использовании эвристических методов процедура поиска скрытых факторов *ресурсоемка*.
* *Параметры эвристических методов необходимо каждый раз подбирать вручную* на пробном наборе данных для достижения наилучшего результата. Чтобы быть уверенным в качестве рекомендательной системы, такую процедуру придется проводить регулярно.

Так как факторы в данном методе являются скрытыми, то невозможно использовать их для обоснования рекомендаций. Даже используя существующие методы по сопоставлению скрытых факторов с явными (описаны подробнее в статье [32]), требуется регулярный пересчет параметров метода, а значит количества и семантики факторов, что делает невозможным их использование для обоснования и фильтрации рекомендаций.

## 1.4 Выводы

Существующие методы основной своей целью ставят предсказание точного значения функции полезности и дальнейшее ее использование для построения рекомендаций. Это вносит ограничение в функциональную часть рекомендательных систем. В частности все рассмотренные методы по построению рекомендательных систем не имеют возможности обоснования рекомендаций из-за отсутствия явных факторов, доступных для понимания пользователей, что в свою очередь не позволяет составить критерии для фильтрации рекомендаций. Для решения этих проблем предлагается переформулировать задачу рекомендательных систем и ограничиться лишь приближенным значением этой функции, формальные требования для которой будут представлены в следующей главе. Так же предлагается выделить из объектов единицу смысла и использовать новое представление данных, а вместе с ним и новый алгоритм для рекомендаций. Это позволит существенно расширить функциональность рекомендательных систем и позволит нескольким рекомендательным системам, построенным по описанному принципу быть совместимыми между собой независимо от конкретной реализации функции полезности объектов.

# Глава 2. Метод вложенных тегов

В данной работе предлагается перейти он неявных факторов к явным и использовать их в сравнении объектов пользователей и объектов, а так же объектов между собой.

## 2.1. Представление данных

Каждую сущность (пользователи и объекты) в системе рекомендаций предлагается представлять как цельную единицу информации, имеющую связи с другими единицами информации. Явными факторами, используемыми для сравнения, будут являться сами единицы информации, которые так же, кроме пользователей и объектов могут быть просто ключевыми словами или характеристиками (которые могут ссылаться на более детальные характеристики, раскрывающие их сущность).

В [23] строится гибридная рекомендательная система, в которой предлагается профиль пользователя представлять в виде онтологий. Такой подход менее динамичен и гибок и требует заранее заданной структуры онтологии и подразумевает иерархию сущностей. В предлагаемом подходе нет ограничений на то, какие объекты часть каких могут являться и в этом смысле представление данных больше по смыслу похоже на граф, чем на онтологию.

Каждую сущность предлагается представлять в виде тега, имеющего ссылки на другие единицы информации, которые могут оказаться пользователями, объектами, произвольными факторами или другими сущностями, которые могу быть введены в рекомендательную систему. Этот принцип представлен на рис. 4. Поиск и сравнение объектов предлагается осуществлять на основе содержимого тега, т.е. набора его ссылок и содержимого тегов по этим ссылкам.

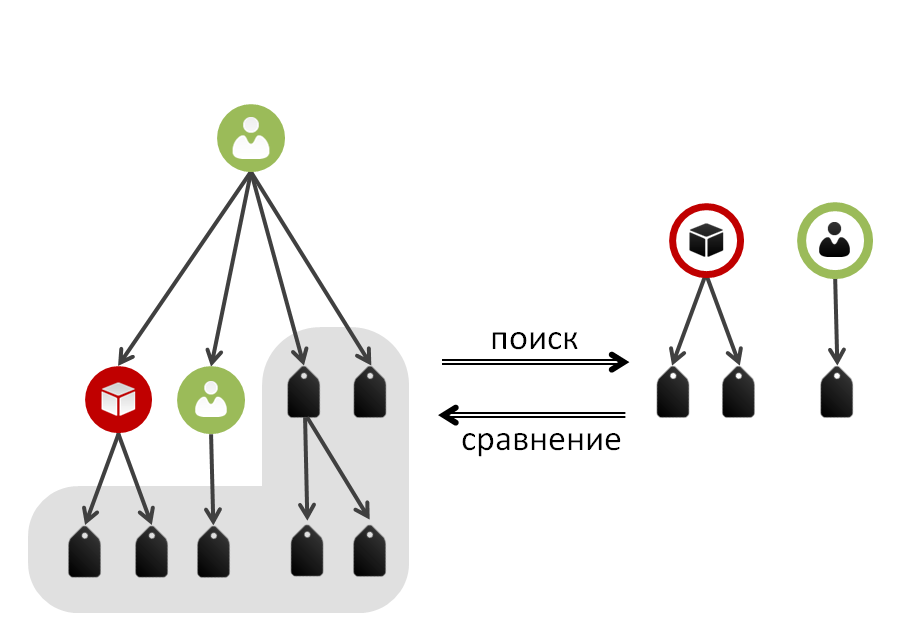


Рис. . Иллюстрация принципа вложенных тегов

Например, тег, соответствующий пользователю Анна, которая увлекается фантастикой, любит книгу Оруэла «1984» и дружит с Борисом, который так же является пользователем рекомендательной системы, будет иметь связи с тегом, соответствующем книге «1984», тегом, соответствующему Борису и тегу, соответствующему жанру фантастики. Тег книги «1984» будет иметь связь с жанровым тегом, соответствующим антиутопиям. Тег пользователя Борис будет иметь связи с тегом книги «Улитка на склоне», который в свою очередь будет иметь связь с авторским тегом, соответствующим писателям Стругацким. Этот пример изображен на рис. 5.

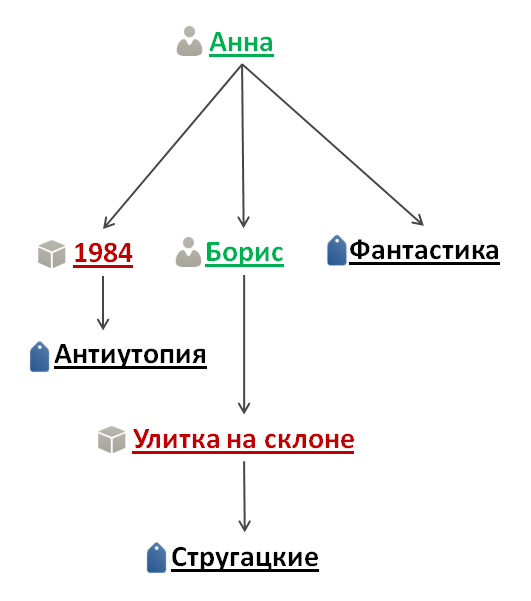


Рис. . Пример вложенного тега

Тогда рекомендательная система среди рекомендуемых пользователю Анна объектов выдаст тег, соответствующий книге «Улитка на склоне» (изображен на рис. 6), потому что эта книга так же является антиутопией, как одна из книг Анны, потому что эта книга тех же авторов, что и книга друга Анны, и потому что эта книга написана в том жанре, который Анна любит. Эти три фактора будут обоснованием того, почему рекомендательная система посоветовала ей эту книгу.



Рис. . Пример вложенного тега для рекомендации пользователю Анна

## 2.2. Сравнение объектов

Сравнение объектов в системе должно происходить на основании их внутреннего содержимого.

За будем обозначать множество тегов, связанных с .

Вес связи между объектами будем обозначать , как это показано на рис. 7.

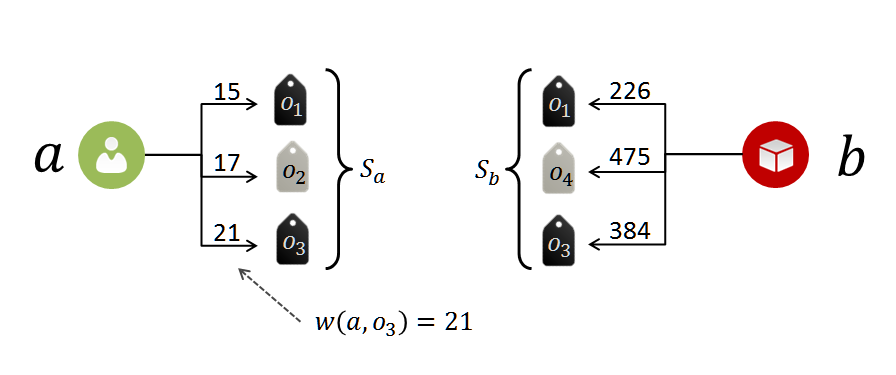


Рис. . Обозначения тегов и весов связей

Для того чтобы иметь возможность обоснования объектов пользователю, сформулируем базовые требования к функции . Не умаляя общности, ограничим значения функции интервалом , подразумевая под этим то, насколько два объекта похожи в процентном отношении. Так же для удобства будем обозначать результат функции в процентах, где 100% будут соответствовать абсолютно идентичным объектам. Для этого относительный вес связи между объектами будем обозначать .

1. Функция должна иметь результат 100% тогда и только тогда, когда множества компонент равны и все веса одинаковы.
2. Для объектов, которые не имеют ни одного общего компонента, функция должна выдавать результат 0%.
3. Во всех остальных случаях результат функции принадлежит открытому интервалу .

Так же сформулируем требования о порядке. Предположим, что имеется три объекта: , и . Так же предположим, что каждый их них связан с объектом и множеством объектов . Иллюстрация этих связей представлена на рис. 1.

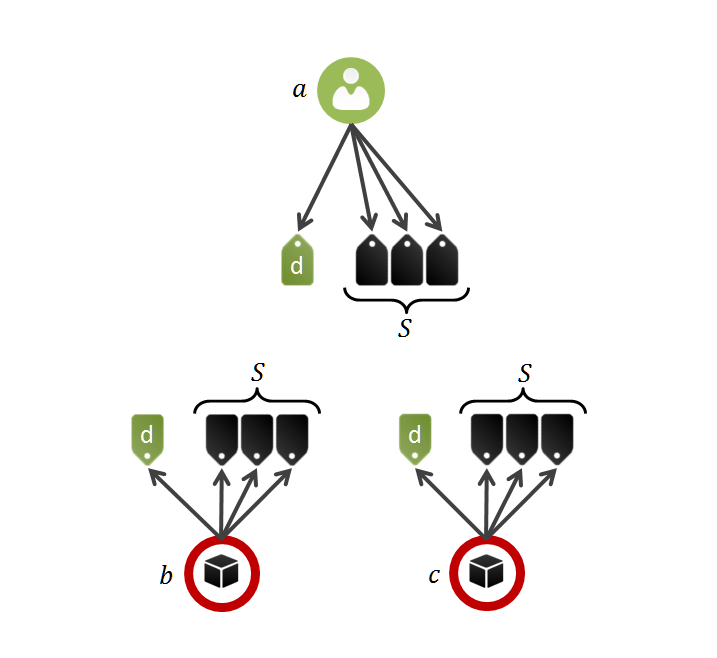
**

Рис. . Иллюстрация примера для описания порядка для функции сравнения с равными подмножествами

Предположим, что у всех трех объектов веса связи с объектами из множества одинаковые.

Тогда, чем ближе вес связи будет ближе к весу , тем больше должно быть значение функции .

Это означает, что чем ближе веса компонентов друг к другу, тем выше значение функции.

Сформулируем требования к детальности описания, пользуясь похожим примером, но со следующей поправкой. На этот раз веса связей с объектом d будут одинаковы, а множества остальных компонент не будут иметь попарных общих пересечений, но будут отличаться мощностью. Иллюстрация этого примера изображена на рис. 8. Тогда:

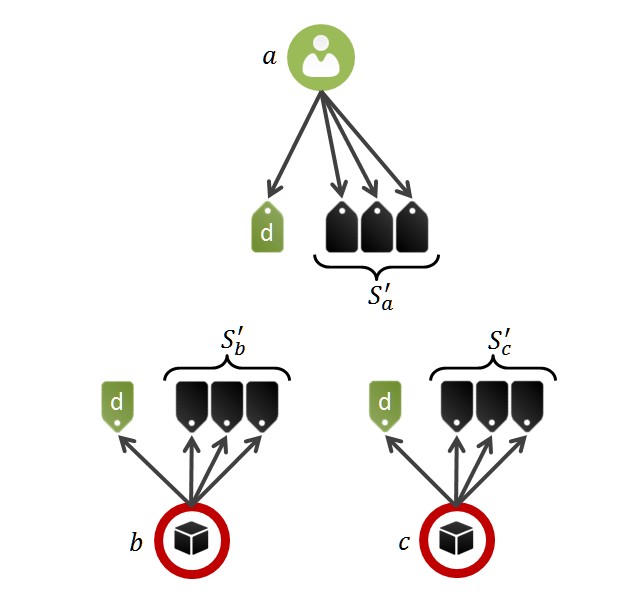
Это значит, что функция полезности будет выше для той пары объектов, где описания близки по детализации.  
**

Рис. . Иллюстрация примера для описания порядка для функции сравнения с подмножествами разных мощностей.

Таким образом, построенная функция будет отражать ожидаемое пользователем поведение, более похожие по описанию объекты будут более похожи между собой, более похожие на пользователя объекты будут рекомендоваться ему раньше других. Заметим, что функция может принимать во внимание большую вложенность объектов, используя для сравнение связи компонентов тега, но при этом вес компонента должен падать с нарастанием глубины. Иллюстрация представлена на рис. 9.

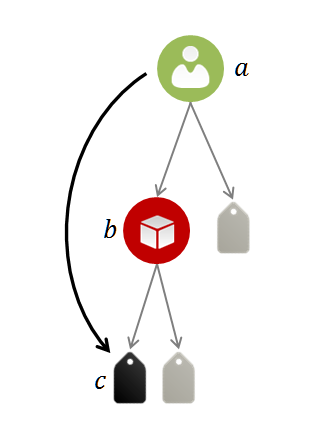


Рис. . Использование вложенности в сравнении объектов.

Таким образом, например, можно учесть данные о структуре объектов, отмеченных пользователем, как это было проиллюстрировано на примере пользователя Анны в разделе 2.1.

Важно заметить, что вес связи должен падать, если это связь с объектом, который является дочерним по отношению к исходному. Иными словами, для примера, приведенного на рис. 9:

При использовании вложенности объектов может возникнуть проблема циклов, если, например Борис добавит в свой профиль Анну, то возникнет цикл из двух тегов Борис-Анна. Решение данной проблемы возлагается на конкретную реализацию функции сравнения.

Приведем пример функции сравнения, не использующей вложенность тегов. Предположим, что сравниваются два объекта и . Обозначим функцию следующим образом:

* Для всех имеющихся связей c другими объектами, эта функция будет эквивалентна функции .
* Для всех остальных связей эта функция будет равна нулю.

Так же обозначим взвешенный вариант функции за .

Тогда предлагаемая функция будет выглядеть следующим образом:

Как видно из формулы, требуемое количество операций для подсчета такой функции будет порядка , где — среднее количество компонент у каждого объекта.

Такая функция будет удовлетворять всем сформулированным требованиям.

Заметим так же, что построенная функция позволяет сравнивать между собой как пользователей, так и объекты, что делает возможным построение гибридной рекомендательной системы с использованием контентных или коллаборативных принципов.

## 2.3. Поиск объектов

Поиск объектов для сравнения, как это было сказано раньше, осуществляется на основе содержимого тега. Процедура поиска рекомендаций состоит из трех основных шагов:

1. Поиск компонентов тега.
2. Поиск тегов, содержащих данный как компонент.
3. Сравнение найденных тегов с исходным.

Иллюстрация этих трех шагов приведена на рис. 10.

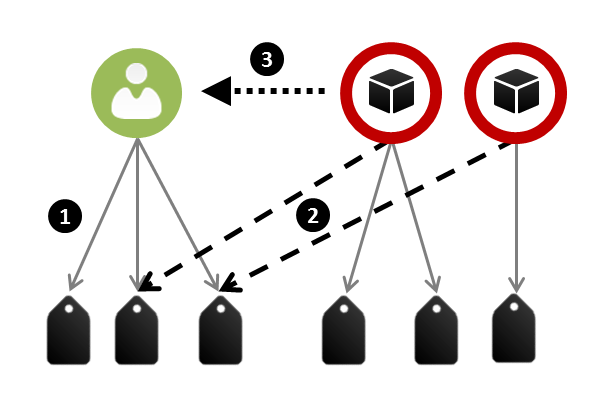


Рис. . Иллюстрация трех основных шагов поиска рекомендаций

Таким образом, будут найдены все объекты, имеющие ненулевое значение функции , так как будут иметь хотя бы один общий компонент с исходным объектом.

Конкретная реализация трех шагов поиска должна зависеть от реализации функции сравнения, чтобы наиболее быстро и точно предоставить объекты, которые будут наиболее релевантными. В данном разделе произведена оценка трудоемкости возможной реализации поиска для функции сравнение, которая не использует вложенность объектов. В главе 3 данной работы, где описано построение экспериментального стенда, находится более подробное описание реализации предложенного метода поиска.

Предположим, что функция сравнения не использует вложенность. Каждый тег в системе будем представлять в виде универсального идентификатора (id). В ассоциативном массиве будем хранить веса связей между объектами (children). В качестве ключа будет выступать идентификатор тега, а в значении элемента массива будет находиться другой ассоциативный массив, ключом которого будет идентификатор элемента, а значением вес связи между этим элементом и исходным. Для ускорения процедуры поиска родителей будем хранить аналогичный массив , в котором величины будут означать вес связи между тегом в массиве и исходным, иначе говоря, списки родителей исходного тега. Иллюстрация структуры массива представлена на рис. 11, структура массива представляется аналогично.

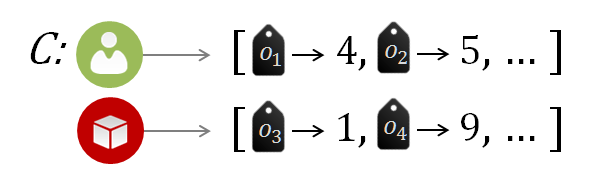
**

Рис. . Структура ассоциативного массива С.

Обозначим объект, к которому ищутся рекомендации за . Тогда первый шаг алгоритма поиска займет одно действие, извлечение списка весов связей по ключу, который является идентификатором объекта .

Следующий шаг алгоритма потребует в среднем действий, если — это среднее количество компонент у объекта. Для каждого компонента из массива будет извлечен список тегов, содержащий данный компонент. Для всех компонентов тега это потребует в среднем порядка действий. Таким образом, будет получен список кандидатов для сравнения с тегом , размер которого будет составлять в среднем . Для осуществления сравнения с тегом потребуется извлечь список компонент для каждого кандидата на сравнение, эта операция потребует действий. При использовании описанной в предыдущем разделе функции сравнение со всеми объектами потребует так же порядка действий. Сортировка полученных результатов может быть выполнена с помощью алгоритма «быстрой сортировки» и потребует порядка действий.

Так как все эти этапы выполняются последовательно, то общий порядок выполненных действий будет .

Порядок занимаемого объема памяти двумя ассоциативными массивами в среднем так же составит .

Из-за преимущества в скорости работы и возможности параметризации поиска такой метод поиска может быть использован в гибридных рекомендательных системах с применением контентных или коллаборативных методик.

# Глава 3. Экспериментальный стенд

В данной главе описывается создание экспериментального стенда для того, чтобы получить данные о возможности использования предложенного метода в промышленных системах. В первом разделе описывается измерение функции предсказания оценки согласно с классическими способами оценки рекомендательных систем. Во втором разделе описано построение экспериментального стенда, демострирующего возможноти обоснования и фильтрации рекомендаций.

## 3.1. Подсчет среднеквадратичной ошибки функции

Разработчики современные рекомендательных систем используют данные существующих пользователей для оценки качества своих систем. Они стремятся предсказать, насколько точно их система угадывает оценку, которую пользователь ставит объекту. Для оценки точности предсказаний чаще всего используется оценка среднеквадратичного отклонения предсказанной оценки от оценки, выставленной пользователем.

### 3.1.1. Используемые данные

В данной работе для измерения точности предсказывания оценки использовались данные MovieLens исследовательской группы GroupLens, которые находятся в открытом доступе в сети интернет по адресу [24].

Для сравнения был выбран набор данных, включающий в себя:

* 10 000 000 рейтингов по пятибалльной шкале.
* 10 681 фильмов
* 100 000 ключевых слов
* 71 567 пользователей

Экспериментальный стенд реализован на языке программирования Java и состоит из следующих компонент:

* *Crawler*, который собирает информацию из данных GroupLens и превращает сохраняет ее с помощью компонента DAO (Data Access Object)
* *Dao*, котрый хранит в себе объекты типа InformationItem (который соответствует вложенному тегу), используя компонент In-memory Storage.
* *In-memory Storage*, который позволяет сохранять и загружать объекты типа InformationItem и хранит их в памяти. Хранение объектов в памяти позволяет существенно сократить время загрузки и сохранения по сравнению с базами данных, спользующих жесткий диск для хранения инфорамции.
* *Recommender*, который осуществляет функции поиска рекомендуемых объектов и подсчет функции . Для загрузки объектов он использует компонент DAO.
* *Benchmark*, который использует Crawler для сбора доступной информации, DAO для загрузки данных и Recommender для сравнения. Этот компонент осуществляет подсчет среднеквадратичного отклонения и передает его компоненту Result.
* *Result*, который выводит результаты пользователю.

Схема компонент показана на рис. 13.

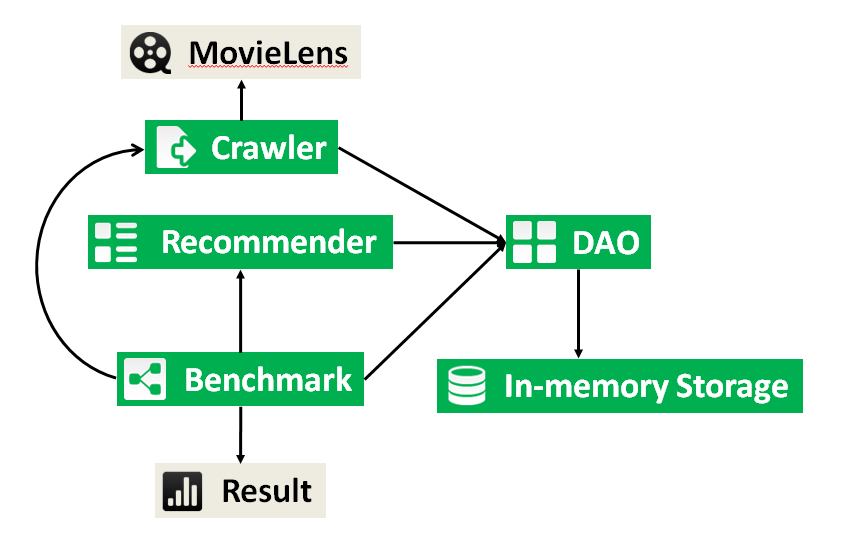


Рис. . Схема компонент экспериментального стенда для измерения среднеквадратичной ошибки.

Используемые данные состоят из трех файлов:

* *movies.dat*, который содержит названия фильмов, их идентификационные номера и сведения о жанрах.
* *ratings.dat*, который содержит данные об оценках пользователей для фильмов.
* *tags.dat*, который содержит данные о том, какие пользователи какие ключевые слова ставили для фильмов.

Файл рейтингов ratings.dat содержит в каждой строке по одному рейтингу пользователя. Он был разделен таким образом, чтобы в файл ratings.tetst попадал каждый пятый рейтинг пользователя, а все остальные в файл ratings.train. Таким образом, был создан обучающий и проверочный набор данных.

Компонент Benchmark собирает сведения о фильмах c помощью компонента Crawler, который использует файлы данных и возвращает данные о фильмах, жанрах, пользователях и тегах. Данные загружаются в следующем порядке:

1. Загружаются данные о фильмах из файла movies.dat. Для каждого нового фильма и жанра создается соответствующий тег, создается связь между фильмом и жанром веса **INITIAL\_TAG\_TO\_MOVIE\_WEIGHT** (см. табл. 1).
2. Загружаются данные о тегах пользоватлей. Для каждого нового ключевого слова и пользователя создается соответствующий тег. Так же создается связь между пользователем и ключевым словом веса **INITIAL\_TAG\_TO\_USER\_WEIGHT** (см. табл. 1). При слудющих использованиях данного тега данным пользователем, вес связи увеличивается на величину **ADDITIONAL\_TAG\_TO\_USER\_WEIGHT** (см. табл. 1).
3. Загружаются данные из файла ratings.train, который содержит в себе рейтинги пользователей по пятибальной шкале. Для каждого рейтинг производятся следующее. Каждый тег найденного фильма добавляется к пользователю с весом **DIFFUSE\_TAG\_TO\_USER\_WEIGHT\_FULL** (см. табл. 1), пропорционально рейтингу фильма. Если такой тег уже имелся у пользователя, то он увеличивается на величину **DIFFUSE\_TAG\_TO\_USER\_WEIGHT\_FULL** (см. табл. 1) пропорционально рейтингу фильма.

|  |  |
| --- | --- |
| **INITIAL\_TAG\_TO\_MOVIE\_WEIGHT** | **1** |
| **ADDITIONAL\_TAG\_TO\_MOVIE\_WEIGHT** | **0.1** |
| **INITIAL\_TAG\_TO\_USER\_WEIGHT** | **1** |
| **ADDITIONAL\_TAG\_TO\_USER\_WEIGHT** | **0.1** |
| **DIFFUSE\_TAG\_TO\_USER\_WEIGHT\_FULL** | **1** |

Табл. . Начальные веса для различных типов данных.

Добавление тегов фильмов к пользователям аналогично добавлению самого фильма, если добавление и подсчет оценок осуществляется только единожды, как это происходит в данном случае. Это позволяет использовать функцию сравнения без использования вложенности.

### 3.1.2. Функция сравнения

Для построения информационного стенда была использована функция сравнения , описанная в разделе 2 главы 2.

### 3.1.3. Предсказание оценки

Для построения отклонения значения функции полезности от тех значений, которые задавали пользователи необходимо привести одни к другим. Функция полезности никак семантически не связана с оценкой объекта пользователем, а предназначена лишь для упорядочивания рекомендуемых объектов между собой. При этом как максимальное значение, так и распределение этой функции сильно варьируется для каждой группы рекомендованных объектов. По этим причинам было принято решение умножать результаты функции на значение параметра **MULTIPLICATOR** для того, чтобы приблизить ее значение к оценкам пользователя. Было проведено несколько экспериментальных замеров с разными значениями этого параметра при использовании коэффициентов из табл. 1. При использовании значения 650 было получено лучшее значение среднеквадратичного отклонения предсказанной оценки от настоящей 1.25.

Используя коэфициенты для импорта данных из табл. 1 результат среднеквадратичного отклонения равен 1.5. Сравнение с другими рекомендательными системами приведено на рис. 1.

Рис. . Сравнение результатов среднеквадратичного отклонения с другими методами.

### 3.1.1. Вывод

Такое значение среднеквадратичного отклонения позволяет использовать разработанный метод в промышленных системах.

## 3.2. Обосновывающая рекомендательная система

В данном разделе описывается построение обосновывающей рекомендательной системы с возможностью фильтрации рекомендаций.

### 3.1.1. Используемые данные

Для построения данного экспериментального стенда были использованы данные MovieLens исследовательской группы GroupLens, которые описаны подробнее в разделе 3.1.1 главы 3. Кроме этого были использованы данные всемирной базы фильмов IMDB [24]. Из всех файлов были использованы следующие:

* actors.list, содержащий список актеров, принимавших участие в съемках фильмов
* actress.list, содержащий список актрис, принимавших участие в съемках фильмов
* keywords.list, содержащий список ключевых слов, описывающих сюжет фильма
* directors.list, содержащий список режиссеров, принимавших участие в съемках фильмов.
* producers.list, содержащий список продюсеров, принимавших участие в съемках фильмов.

Данные о персоналиях из IMDB были добавлены фильмам, которые были найдены в списке фильмов MovieLens.

### 3.1.2. Функция сравнения

Для построения информационного стенда была использована функция сравнения , описанная в разделе 2 главы 2.

### 3.1.3. Архитектура системы

Общая архитектура построенного экспериментального стенда представлена на рис. 14.

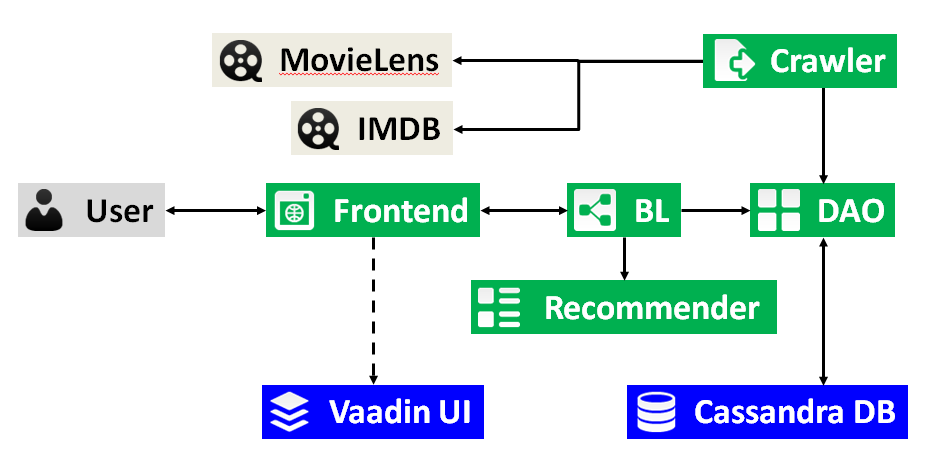


Рис. . Архитектура экспериментального стенда обосновывающей рекомендательной системы с возможностью фильтрации рекомендаций.

Экспериментальный стенд представляет собой веб-приложение, написанное на языке Java, которое предполагает быть запущенным внутри сервера приложений Java, и быть доступным пользователю через браузер. Компоненты системы выполняют следующие функции:

* *Crawler*, который собирает информацию из данных GroupLens и превращает сохраняет ее с помощью компонента DAO (Data Access Object)
* *Dao*, который хранит в себе объекты типа InformationItem (который соответствует вложенному тегу), используя компонент In-memory Storage.
* *Cassandra DB* является NoSQL базой данных, разработанной компанией Facebook Inc., которая используется в данной работе для хранения связей между объектами. Более подробную информацию можно найти на сайте [25].
* *BL (Business Logic)* является компонентом, реализующимлогику java приложения, обрабатывающего запросы компонента Frontend и возвращающее необходимые объекты, содержащие в себе вложенные теги.
* *Frontend* является компонентом, генерирующим страницы для отображения в барузере пользователя, который реализован с помощью технологии Vaadin.
* *Vaadin UI* является java-фреймворком, позволяющим описывать макеты страниц, производить их сборку и отправку в качестве результатов обработки http запросов пользователя [26].

Каждый компонент описан в виде интерфейса и имеет реализацию. Это делает слабыми связи между компонентами, позволяя легко заменять реализации любого узла или использовать существующие реализации для построения другой рекомендательной системы, основанной на методе вложенных тегов.

Гибкость метода вложенных тегов позволяет расширять набор используемых сущностей в системе. Например, использование взимообмена элементами позволит решить проблемы холодного старта и недостаточно детального описания объектов (рис. 16). Система может добавлять пользователю в профиль актеров, режиссеров и ключевые слова фильмов с некоторым начальным весом, если он отмечает фильм как понравившийся и не добавлять сам фильм (отмечено 1 на рис. 16). Фильму так же могут добавляться ключевые слова, добавленные пользователем себе в профиль, что позволит собрать более исчерпывающие данные о фильме.

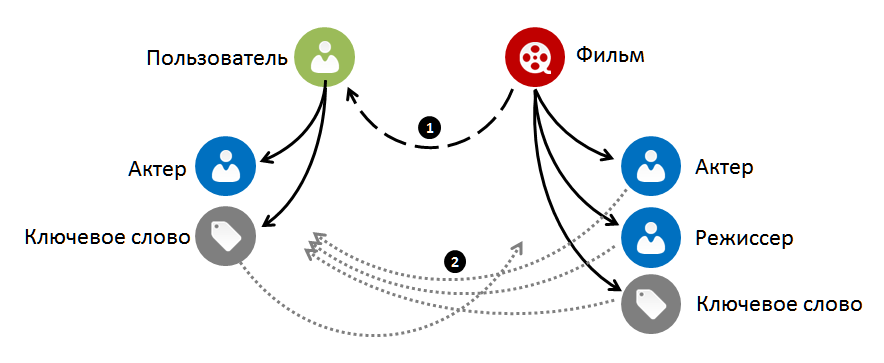


Рис. . Взаимообмен тегами.

### 3.1.4. Внешний вид системы и элементы управления

В построенном экспериментальном стенде используется два основных типа страниц:

* Страница с просмотром информации о теге (рис. 14).
* Страница с фильтрацией рекомендаций (рис. 15).

Построенная система не различает такие сущности объектов как объект или пользователь, поэтому просмотр рекомендаций осуществляется на странице А. В левой части страницы отображено название вложенного тега. В случае пользователя, актера или режиссера в качестве названия используется его имя, а в случае ключевых слов сами слова.

На примере продемонстрирован профиль автора работы с добавленными вручную тегами в профиль пользователя. Автор подбирал теги исходя из тех фильмов, которые ему нравятся, акцентируя внимание на гангстерских фильмах и фильмах про космические сражения. Как видно на рис. 14, система порекомендовала автору фильмы про гангстеров и космические сражения вперемешку. В качестве демонстрации возможности рекомендации неоднородных объектов в систему был добавлен другой пользователь со схожими интересами, которого система так же выдала в качестве рекомендаций.

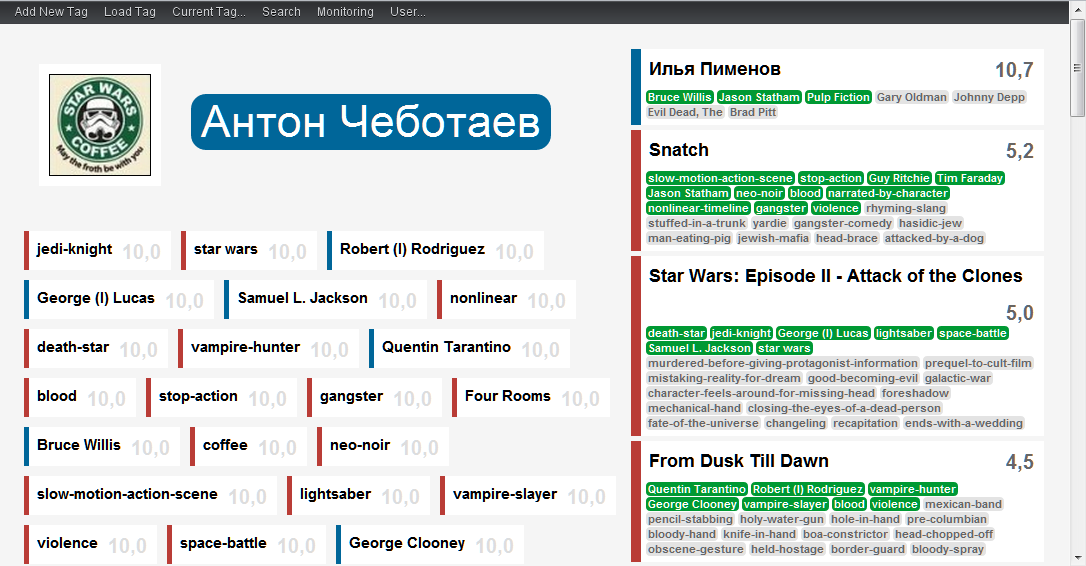


Рис. . Внешний вид страницы просмотра вложенного тега.

Так же слева показаны связи отображаемого тега с другими тегами. Каждая связь представлена в виде прямоугольника, в котором вписаны название тега и вес связи.

Рекомендованные объекты отображаются в столбце справа и упорядочены по значению функции . Теги, связи с которыми присутствуют у рекомендованных объектов и у отображаемого тега выделены зеленым фоном, что позволяет их легко заметить рядом с другими тегами.

Таки обоснования позволяют легко понять, почему теги были порекомендованы в качестве похожих на отображаемый тег, и выбрать среди них интересующий пользователя в данный момент. Меню сверху имеет кнопку добавления текущего отображаемого тега в профиль пользователя, пользующегося системой в данный момент, что позволят быстро наполнять профиль.

Страница с фильтрацией рекомендаций (рис. 15) состоит из поля для добавления вложенных тегов, которые будут использованы в качестве критериев поиска. Имеется возможность добавления тега по имени или уникальному ключу (UUID). Система формирует новый вложенный тег, к которому добавляет связи всех тегов, использованных в качестве запроса, и их компонент, и возвращает похожие на созданный теги в качестве результатов поиска.

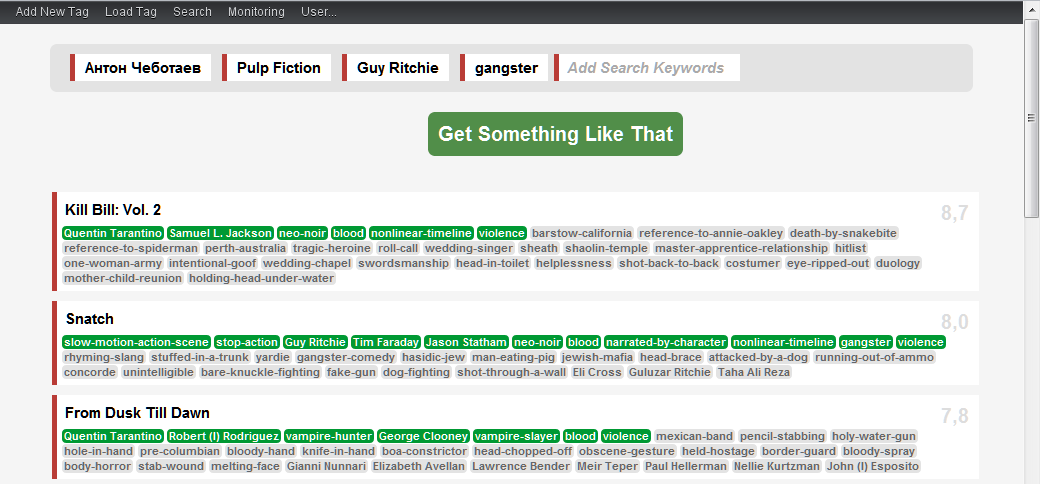


Рис. . Внешний вид страницы поиска и фильтрации рекомендаций.

Так, используя профиль пользователя в качестве одного из критериев запроса и фильм в качестве другого, можно отфильтровать рекомендации и выбрать фильмы, подходящие пользователю и похожие на выбранный фильм одновременно. В предыдущем примере система рекомендовала автору смотреть фильмы про гангстеров и космические сражения одновременно. На рис. 17 показано, как используя параметры, можно отфильтровать рекомендации и получить только список гангстерских боевиков.

### 3.1.1. Вывод

Построен экспериментальный стенд рекомендательной системы по предложенному методу, доказывающий возможность построения обосновывающей рекомендательной системы на основе метода вложенных тегов.

Реализация экспериментального стенда построена с разбиением на логически слабо связанные компоненты, что позволяет использовать их в промышленных системах для внедрения функции рекомендации объектов.

# Заключение

*«Растёт понимание того, что хорошая точность рекомендаций сама по себе не удовлетворяет потребности пользователей РС и не характеризует эффективность её работы».* [48]

По изученным в главе 1 материалам и проделанной исследовательской работе можно заключить, что методы построения рекомендательных систем без явных факторов не позволяют строить обосновывающие РС. В данной работе был предложен метод, позволяющий строить подобные системы, способные работать при меньших затратах ресурсов.

К недостаткам предложенного метода можно отнести зависимость от собранных данных, что является недостатком, присущим любой рекомендательной системе.

В работе был построен экспериментальный стенд, показавший, что построенный метод уступает в точности предсказываемой оценки существующим, но обладает приемлемой точностью для использования в промышленных системах.

Научная новизна работы состоит в том, что рассмотренные в работе принципы сравнения и поиска могут быть использованы не только как часть самостоятельного метода построения рекомендательных систем, но и как составляющая часть гибридных рекомендательных систем.

*«Может быть, самые лучшие алгоритмы должны оцениваться по тому, насколько хорошо они могут представить пользователям обоснование вынесенного решения».* [48]

# Источники

1. *George A. Akerlof.* The Market for "Lemons": Quality Uncertainty and the Market Mechanism // The Quarterly Journal of Economics, v.84, August 1970, p.488-500
2. *J. Herlocker, J. Konstan, L. Terveen, and J. Riedl.* Evaluating collaborative filtering recommender systems // ACM Translations on Information Systems, Vol. 22(1), 200
3. *G. Linden, B. Smith, and J. York.* “Amazon.com Recommendations: Item-to-Item Collaborative Filtering,” IEEE Internet Computing, Jan./Feb. 2003.
4. *B.N. Miller, I. Albert, S.K. Lam, J.A. Konstan, and J. Riedl.* “MovieLens Unplugged: Experiences with an Occasionally Connected Recommender System,” Proc. Int’l Conf. Intelligent User Interfaces, 2003.
5. *D. Billsus, C.A. Brunk, C. Evans, B. Gladish, and M. Pazzani.* “Adaptive Interfaces for Ubiquitous Web Access,” Comm. ACM, vol. 45, no. 5, pp. 34-38, 2002.
6. *Gediminas Adomavicius, and Alexander Tuzhilin.* “Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions”, IEEE Transactions on Knowledge and Data Engineering, Vol. 17, No. 6, 2005
7. *Robin van M. and Maarten van S.* Using Content-Based Filtering for Recommendation // NetlinQ Group, Gerard Brandtstraat 26-28, 1054 JK, Amsterdam, The Netherlands
8. *R. Baeza-Yates and B. Ribeiro-Neto.* Modern Information Retrieval // Addison-Wesley, 1999.
9. *G. Salton.* Automatic Text Processing. Addison-Wesley, 1989.
10. *N. Belkin and B. Croft.* “Information Filtering and Information Retrieval,” Comm. ACM, vol. 35, no. 12, pp. 29-37, 1992.
11. *Yehuda Kore, Robert Bell and Chris Volinsky.* “Matrix Factorization Techniques for Recommender Systems” Published by the IEEE Computer Society 0018-9162/09/$26.00 © 2009 IEEE
12. *Anne Yun-An Chen, Dennis McLeod*. «Collaborative Filtering for Information Recommendation Systems».
13. *M. Balabanovic and Y. Shoham.* “Fab: Content-Based, Collaborative Recommendation,” Comm. ACM, vol. 40, no. 3, pp. 66-72,1997.
14. *N. Good, J.B. Schafer, J.A. Konstan, A. Borchers, B. Sarwar, J.L.Herlocker, and J. Riedl*. “Combining Collaborative Filtering with Personal Agents for Better Recommendations,” Proc. Conf. Am.Assoc. Artificial Intelligence (AAAI-99), pp. 439-446, July 1999.
15. *J.S. Breese, D. Heckerman, and C. Kadie*. “Empirical Analysis of Predictive Algorithms for Collaborative Filtering,” Proc. 14th Conf. Uncertainty in Artificial Intelligence, July 1998.
16. *J. Delgado and N. Ishii*. “Memory-Based Weighted-MajorityPrediction for Recommender Systems,” Proc. ACM SIGIR ’99Workshop Recommender Systems: Algorithms and Evaluation, 1999.
17. *A. Nakamura and N. Abe*. “Collaborative Filtering UsingWeighted Majority Prediction Algorithms,” Proc. 15th Int’l Conf. Machine Learning, 1998.
18. *D. Billsus and M. Pazzani*. “Learning Collaborative Information Filters,” Proc. Int’l Conf. Machine Learning, 1998.
19. *L. Getoor and M. Sahami*. “Using Probabilistic Relational Modelsfor Collaborative Filtering,” Proc. Workshop Web Usage Analysis and User Profiling (WEBKDD ’99), Aug. 1999.
20. *K. Goldberg, T. Roeder, D. Gupta, and C. Perkins*. “Eigentaste: A Constant Time Collaborative Filtering Algorithm,” Information Retrieval J., vol. 4, no. 2, pp. 133-151, July 2001.
21. *T. Hofmann* “Collaborative Filtering via Gaussian Probabilistic Latent Semantic Analysis,” Proc. 26th Ann. Int’l ACM SIGIR Conf.,2003.
22. *J. Ben Schafer and Joseph A. Konstan and John Riedl.* Meta-recommendation Systems: User-controlled Integration of Diverse Recommendations
23. *Stuart E. Middleton, Nigel R. Shadbolt, and David C. de Rouge* Ontological User Profiling in Recommender Systems
24. GroupLens MovieLens data <http://www.grouplens.org/node/73>
25. International Movie Database <http://www.imdb.com/interfaces>
26. База данных Cassandra <http://cassandra.apache.org/>
27. Java web-framework Vaadin [http://vaadin.com](http://vaadin.com/)