1. ЛЕКЦИЯ. Рекомендательные системы

Рекомендательная система подбирает и предлагает пользователю, релевантный контент, основываясь на своих знаниях о пользователе, контенте и взаимодействии пользователя и контента. Рекомендательная система стремится подобрать и предложить контент, который пользователь ещё не видел и который наиболее вероятно будет ему интересен, то есть это система прогнозирования предпочтений. Рекомендательные системы представляют собой программный комплекс, который определяет и распознает интересы и предпочтения пользователей, формируя рекомендации в соответствии с ними.

Дадим несколько определений, чтобы убедится, что мы правильно понимаем друг друга (табл. 1).

Таблица 1.

Термин	Определение
Прогноз	Прогноз – это предположение, относительно того, насколько пользователю понравится контент.
Релевантность	Расположение контента в соответствии с тем, что всего подходит пользователю в данный момент. Релевантность сочетает в себе контекст, демографические данные и (ожидаемые) оценки.
Рекомендация	Лидеры по релевантности
Персонализация	Сочетание релевантности и наглядности

Рекомендаторы используют данные о контенте, о пользователях и о взаимодействиях пользователей и контента. Такие рекомендации являются персональными, так как основаны на персональных предпочтениях и формируются специально для данного пользователя. В отличии от, например, рекомендации популярных или новых товаров.

Самые первые рекомендательные системы появились в конце 1990-х. Это был первый цифровой видеомагнитофон TiVo *(TuBo)*, который пытался учесть предпочтения пользователей. Устройство продавалось конечным пользователям такими брендами, как Sony и Philips. В то время хорошо структурированных описаний для всех вещаемых телепередач не было и мощности пользовательских устройств для сложных вычислений тоже не хватало, поэтому в основу системы рекомендаций был положен алгоритм коллаборативной фильтрации.

Коллаборативная фильтрация основывается исключительно на сходстве поведения пользователей и не использует никаких описаний телепередач и фильмов или другой семантики. Если на двух устройствах цифровых

видеомагнитофонах в двух разных домохозяйствах зрители обычно выбирали одни и те же программы, а потом первый пользователь начинал регулярно смотреть какую-либо новую телепередачу, то эту телепередачу рекомендовали и второму пользователю. Все видеомагнитофоны с данной функцией отправляли на центральный сервер данные о последовательностях кликов (какие именно кнопки нажимал пользователь, какие программы смотрел и записывал), на основании этих данных рекомендательный алгоритм на сервере вычислял персональные рекомендации, и они затем загружались на устройства. Такая организация процесса на центральном сервере позволяла, во-первых, сравнивать и находить зрителей с похожими вкусами, а во-вторых, не «грузить» видеомагнитофоны TiVo *(TuBo)* с их очень ограниченными вычислительными возможностями. Несмотря на то, что решение не работало в реальном времени и иногда неправильно определяло интересы пользователя, простота подхода быстро сделала его популярным в отрасли.

Чемпионом применения стала компания Netflix *(нетфлекс)*, которая для своего бизнеса предоставления DVD-дисков в аренду привлекла команду разработчиков, настраивающих параметры рекомендательного движка коллаборативной фильтрации, и добивался год от года все лучших результатов.

Однако в середине 2000-х качество рекомендаций, которое может обеспечить алгоритм коллаборативной фильтрации, достигло некоторого предела. В надежде спасти ситуацию за счет коллективного разума в 2006 Netflix *(нетфлекс)* организовала конкурс Netflix Prize *(нетфлекс прайз)*. Лучшие команды разработчиков в течение трех лет добились дальнейшего улучшения точности рекомендаций еще на 10%, но предложенные решения использовать на практике было нельзя – небольшое улучшение точности требовало совершенно неадекватных операционных затрат. Коллаборативная фильтрация как основной метод рекомендаций в видео зашла в тупик.

В процессе развития новых подходов к решению задачи на гребне волны оказались новые компании, в частности Jinni *(джини)* и Aprico *(эйприко)*, которые поняли, что качественная информация о рекомендуемом контенте могла бы служить серьезной основой для тонкой подстройки под интересы зрителей. Эта новая волна исследований в основном использовала байесовскую классификацию, которая позволяет понять и оценить, почему пользователь предпочитает тот или иной контент.

В Netflix *(нетфлекс)* сразу оценили перспективы нового направления семантического анализа описаний и снова решили заняться собственной разработкой. Они стали создавать описания и классификаторы для фильмов

своего каталога. Этот подход имеет ряд проблем, связанных с масштабированием, так как разумен и оправдан, только если каталог видео относительно мал и увеличивается медленно. Остальные компании, занимающиеся семантическими рекомендательными решениями, сфокусировались на классификации контента на основе метаданных, которые существовали на рынке. Netflix *(нетфлекс)* же опять отличился и постарался дойти до границ возможного. Его каталог жанров быстро вырос с исходных 560 вариантов до 93116, по данным на июль 2014 года, в числе которых были, например, такие как «эмоциональные драмы 80-х», куда в каталоге Netflix *(нетфлекс)* попадало лишь 2 фильма. И это опять было чересчур – категории, которым соответствуют один-два объекта, не годятся для анализа и становятся незначимыми для байесовского классификатора.

Таким образом, в базовых подходах для рекомендательных систем могут использоваться два вида данных:

1) Информация о взаимодействии пользователей с объектами интереса;

2) Информация, предоставленная самими пользователями, например, атрибуты, указанные в профиле или релевантные ключевые слова.

Первую группа методов чаще всего называют методами коллаборативной фильтрации, для методов второй группы обычно используется название рекомендаций на основе контента (рис. 1).

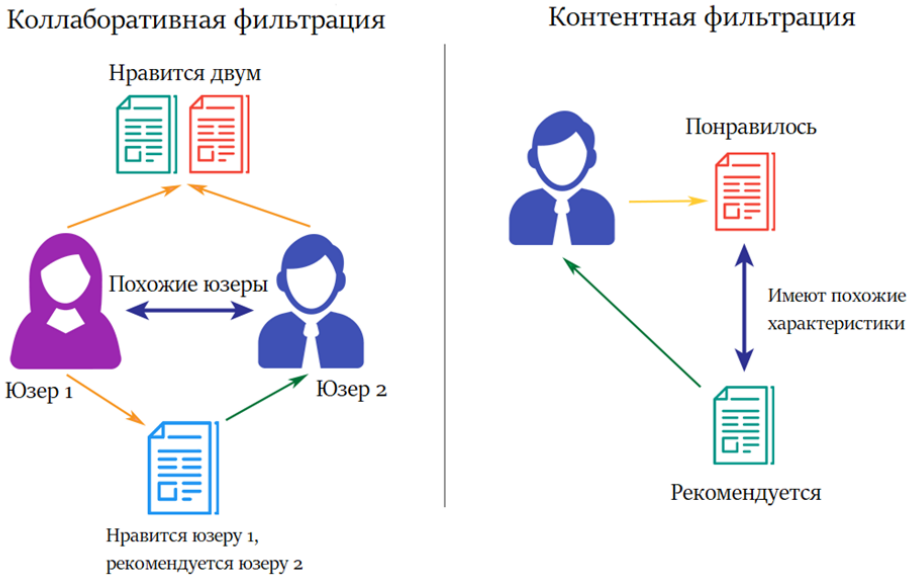

Рис. 1. Иллюстрация двух базовых подходов.

Еще один тип систем рекомендаций – системы рекомендаций, основанные на знаниях: здесь рекомендации основаны на явно указанных пользовательских требованиях. Некоторые рекомендательные системы могут

объединять перечисленные выше аспекты; такие системы называются гибридными. Они сочетают в себе сильные стороны различных подходов для создания методов, которые могут работать более эффективно в узкопрофильных системах.

Рекомендательные системы уже интегрированы во множество веб-приложений, которые широко используются каждый день миллионами пользователей. Рассмотрим примеры крупнейших ресурсов, использующие рекомендательные механизмы.

LinkedIn *(лентейн)* – бизнес-ориентированная социальная сеть. Встроенный рекомендательный механизм предлагает пользователю рекомендации людей, которых он, возможно, знает, вакансий, которые могли бы его привлечь, групп, в которые он мог бы захотеть вступить, компаний, которыми он мог бы заинтересоваться. Специализированная система коллаборативной фильтрации LinkedIn *(лентейн)* основана на технологии Apache Hadoop *(апачи хадип)* – свободно распространяемый набор утилит, библиотек и фреймворк для разработки и выполнения распределённых программ, работающих на кластерах из сотен и тысяч узлов.

Amazon *(амазон)* – одна из крупнейших площадок интернет-торговли – использует рекомендации на основе контента. Когда посетитель выбирает для покупки какой-либо товар, Amazon *(амазон)* на основе этого исходного товара рекомендует посетителю другие товары, приобретенные другими пользователями (с помощью матрицы покупки следующего товара на основе его схожести с предыдущей покупкой). Компания Amazon *(амазон)* запатентовала этот подход под названием «коллаборативная фильтрация от элемента к элементу».

Last.fm (*ласт. фм*) создает музыкальную «станцию» рекомендованных песен, наблюдая, какие группы и отдельные треки пользователь прослушивает на регулярной основе. Last.fm (*ласт. фм*) воспроизводит дорожки, которые не присутствуют в библиотеке пользователя, но часто воспроизводятся другими пользователями с аналогичными интересами. Поскольку этот подход использует поведение пользователей, он является примером совместной фильтрации.

Pandora (*пандора*) использует метаданные песен и исполнителей порядка 400 атрибутов, предоставленных проектом Music Genome Project, чтобы сгенерировать «станцию», которая воспроизводит музыку с похожими свойствами. Кроме того, для уточнения результатов «станции» используется обратная связь от пользователя, которая обесценивает определенные атрибуты,

когда пользователю не понравилась определенная песня и увеличивает вклад других атрибутов, когда пользователю нравится песня. Данный сервис использует контент-ориентированный подход.

Постановка задачи

Для начала формализуем нашу задачу. Имеются данные:

- множество пользователях $(users, u \in U),;$
- множество объектов $(items, i \in I)$ (фильмы, треки, товары и т.п.);
- событиях $(events, (u, i, r_{ui}) \in D)$ (действия, которые пользователи совершают с объектами)

Событие описывается так: пользователь u поставил оценку r_{ui} объекту i.

Требуется:

- предсказать оценку объекту, которого пользователь ещё не видел:

$$r_{ui} = Predict(u, i)$$

- вычислить персональные рекомендации для пользователя u:

$$u \longmapsto (i_1, \dots, i_k) = Recommend_K(u)$$

- похожие объекты:

$$u \longmapsto (i_1, \dots, i_k) = Similar_M(i).$$

Нужно как можно лучше предсказать, какие оценки должны быть в ячейках со знаками вопроса (рис.2)

	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6
User 1	5	4	5			
User 2	4	?	5			
User 3		3	5	?	4	
User 4			?	3	4	
User 5	?		4	2	4	?
User 6	3					5

Рис. 2. Пример имеющихся данных об оценках.

Классификация рекомендательных систем

В базовых подходах для рекомендательных систем могут использоваться два вида данных:

- Информация о взаимодействии пользователей с объектами интереса.
- Информация, предоставленная самими пользователями, например, атрибуты, указанные в профиле или релевантные ключевые слова.

Первую группа методов чаще всего называют методами коллаборативной фильтрации, для методов второй группы обычно используется название

рекомендаций на основе контента. Некоторые рекомендательные системы могут объединять перечисленные выше аспекты; такие системы называются гибридными. Они сочетают в себе сильные стороны различных подходов для создания методов, которые могут работать более эффективно в узкопрофильных системах (рис.2).

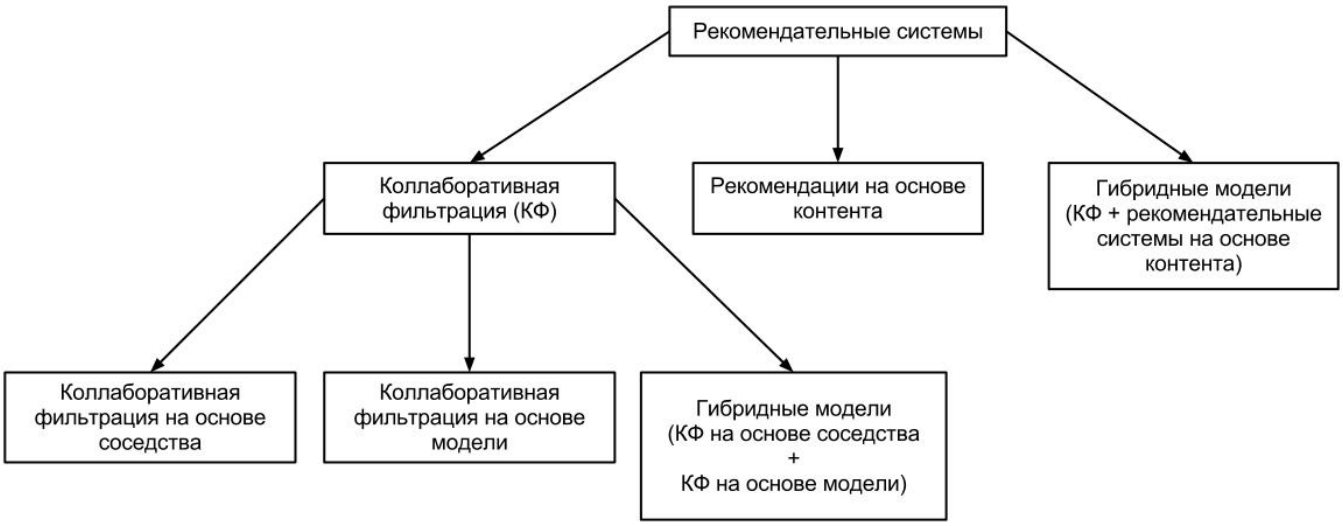

Рис. 2. Классификация рекомендательных систем

Коллаборативная фильтрация, в свою очередь, также разделяется на 3 основных подхода (типа).

1. Рекомендательные системы на основе коллаборативной фильтрации

Коллаборативная фильтрация, совместная фильтрация (*англ. Collaborative filtering*) вырабатывает рекомендации, основанные на модели предшествующего поведения пользователя. Эта модель может быть построена исключительно на основе поведения данного пользователя или — что более эффективно — с учетом поведения других пользователей со сходными характеристиками.

Основная идея алгоритмов коллаборативной фильтрации заключается в предложении новых элементов для конкретного пользователя на основе предыдущих предпочтениях пользователя или мнения других единомышленников пользователя. На сегодняшний день исследователи разработали целый ряд алгоритмов коллаборативной фильтрации, которые можно разделить на две основные категории:

1. Методы, основанные на анализе имеющихся оценок, – анамнестические методы (Memory-based) (анамнестический, анамнез – совокупность медицинских сведений, получаемых путем опроса, обследуемого и знающих его лиц). Эти алгоритмы основываются на статистических методах,

чтобы найти группу пользователей близких к целевому пользователю. Этот подход еще называют метод ближайших соседей: использование предшествующих оценок, сделанных клиентом, и анализ оценок других пользователей, которые имеют подобные предпочтения. Тогда рекомендации (прогноз) для целевого пользователя формируются на основании вычисления некой меры похожести по всем накопленным данным (рис.3).

Рис. 3. Методы коллаборативной фильтрации

2. Методы, основанные на анализе модели данных, – модельные методы (Model-based). В этом случае сначала по совокупности оценок формируется описательная модель предпочтений пользователей, товаров и взаимосвязи между ними, а затем формируются рекомендации на основании полученной модели. Процесс формирования рекомендаций разбит на два этапа:

ресурсоемкое обучение модели в отложенном режиме и достаточно простое вычисление рекомендаций на основе существующей модели в реальном времени. Эти алгоритмы могут быть основаны на вероятностном подходе, кластерном анализе, анализе скрытых факторов.

3. Методы, основанные на объединении предыдущих алгоритмов, – *гибридные методы.*

1.1. Анамнестические методы (на основе соседства, окрестности)

Фильтрация в окрестности может быть реализована двумя методами: user-based *(юзер-бейсд – сходства пользователей)* и item-based *(айтем-бейсд – сходства элементов)*, они основаны на построении матриц схожести.

В общем задача нахождения схожести может быть определена следующим образом: имеется два элемента i_1 и i_2; сходство между ними определяются функцией $sim(i_1, i_2)$. Возвращаемое этой функцией значение пропорционально степени сходства между элементами. Тогда для идентичных элементов $sim(i_1, i_2) = 1$, а для элементов не имеющих ничего общего $sim(i_1, i_2) = 0$. Изменение сходства тесно связано с расчетом различия между элементами. Математически это можно выразить так: $\text{Сходство} = 1 - \text{Различие}$.

Целью обоих направлений user-based и item-based является выделение схожих объектов в группы на основе матрицы оценок. В первом случае определяется сходство пользователей: найти других пользователей, чьи прошлые оценки поведения похожи на те, что и у текущего пользователя, и использовать их оценки других элементов для прогнозирования предпочтения текущего пользователя. Второй подход, на основе сходства элементов, в этом случае вместо того чтобы использовать подобие между поведением пользовательских оценок для прогнозирования предпочтения, используется сходство между оценками моделей элементов. Если два элемента, как правило, имеют одинаковые оценки пользователей, то они похожи, и пользователи должны иметь аналогичные предпочтения для подобных элементов.

Для определения сходства между пользователями или элементами используют различные подходы.

Расстояние Жаккара.

Этот параметр называется коэффициентом сходства Жаккара, который показывает на сколько похожи два набора данных. Коэффициент Жаккара измеряет подобие между конечными множествами выборок, и определяется как размер пересечения, деленного на размере объединения множеств выборок.

$$sim\,(a,b) = {|A \cap B|}/{|A \cup B|}$$

Наборы данных можно получить из пользовательских покупок, превращенных в список. В каждой строке списка указано купил – 1 или не купил – 0 пользователь товар. Таким образом создается бинарная таблица 1.

Таблица 1

	О1	О2	О3	О4	О5	О6
П1	1	1	0	0	0	0
П2	1	1	1	0	0	0
П3	1	0	0	0	0	0
П4	0	1	0	1	0	0
П5	0	0	0	0	1	1
П6	0	0	0	0	1	1

Примечание: О – объект, П – покупатель.

Выберем два объекта О1 и О2 (табл.2).

Таблица 2

	О1	О2	Сумма
П1	1	1	1
П2	1	1	1
П3	1	0	0
П4	0	1	0
П5	0	0	1
П6	0	0	1
			4

Чтобы вычислить сходство между двумя объектами нужно подсчитать количество одинаковых битов, т.е. сколько было совершено одинаковых действий. В табл. 2 в четырех строках из шести покупатели сделали тоже самое, т.е. сходство Жаккара между объектами составляет $4/6 = 2/3$.

Также можно найти сходство между объектами купивших товар, нужно подсчитать сколько пользователей купили оба объекта, а затем разделить на количество клиентов, купивших один и оба товара (общее количество покупателей). Тогда сходство Жаккара для покупателей равно $2/4 = 1/2$. Аналогично, можно определить сходство Жаккара для людей, не купивших объект (равно $2/4 = 1/2$).

Рассчитанные показатели достаточно высокие, но лучше попробовать разные функции сходства и оценить какая лучше подходит для решаемой задачи.

Измерение расстояния с помощью L_p-норм.

Норма – функционал, заданный на векторном пространстве и обобщающий понятие длины вектора или абсолютного значения числа. Общая формула для $L_p\ norm$:

$$\|x\|_p = \left(\sum_i |x_i|^p\right)^{1/p}$$

L_1-норма.

Расстояние L_1 также известно, как расстояние городских кварталов, манхэттенское расстояние, расстояние такси, метрика прямоугольного города — оно измеряет дистанцию не по кратчайшей прямой, а по блокам. Название «манхэттенское расстояние» связано с уличной планировкой Манхэттена.

$$\|x\|_1 = \sum_i |x_i|$$

Расстояние городских кварталов $\|x\|_1$ между двумя векторами p, q в -мерном вещественном векторном пространстве с заданной системой координат – сумма длин проекций отрезка между точками на оси координат. Более формально,

$$\|x\|_1(p,q) = \|p-q\|_1 = \sum_{i=1}^{n} |p_i - q_i|$$

где $p = (p_1, p_2, \dots, p_n)$ и $q = (q_1, q_2, \dots, q_n)$ – векторы.

На плоскости расстояние городских кварталов между (p_1, p_2) и (q_1, q_2) равно $|p_1 - q_1| + |p_2 - q_2|$.

Рассмотрим пример определения сходства согласно расстояние городских кварталов (рис.4).

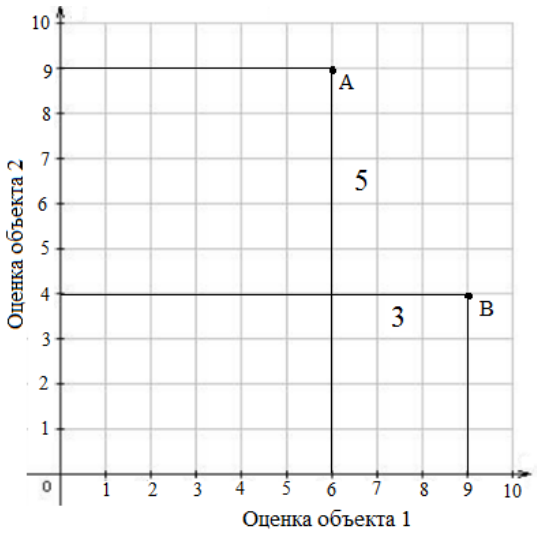

Рис. 4. Сходство двух клиентов на основе оценок двух объектов

Пусть имеются два объекта (объект 1 и объект 2) и два клиента (А и В). Необходимо оценить объекты по шкале от 1 до 10. Клиент А поставил по объекту 1 оценку 6, а вот по объекту 2 поставил – 9. Клиент В поставил по объекту 1 оценку 9, а по объекту 2 поставил – 4.

На основе оценок по объектам

$$\|x\|_1(a,b) = \sum_{i=1}^{n} |a_i - b_i| = |a_1 - b_1| + |a_2 - b_2| = |9-4| + |9-6| = 8$$

Таким образом, сходство по L_1-норме равно 8.

L_2-норма.

L_2-норму иначе называется евклидовым расстоянием. Евклидова метрика (евклидово расстояние) – метрика в евклидовом пространстве – расстояние между двумя точками евклидова пространства, вычисляемое по теореме Пифагора.

Для векторов $p = (p_1, p_2, \dots, p_n)$ и $q = (q_1, q_2, \dots, q_n)$ евклидово расстояние определяется следующим образом:

$$\|x\|_2(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$$

Евклидова метрика — наиболее естественная функция расстояния, возникающая в геометрии, отражающая интуитивные свойства расстояния между точками.

Рассчитаем евклидово расстояние для рис.4.

$$\|x\|_2(a,b) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2} = (9-4)^2 + (9-6)^2 = \sqrt{34} = 5{,}83$$

Таким образом, сходство по L_2-норме равно 5,83.

Коэффициент Отиаи.

Коэффициент Отиаи (косинусный коэффициент, косинусное подобие) – бинарная мера сходства, предложенная японским биологом Акирой Отиаи. Косинусный коэффициент — мера подобия между двумя массивами данных, вычисляемая как косинус угла между векторами в многомерном пространстве (рис. 5). В самом деле, двух пользователей разумно считать похожими, если угол между их векторами предпочтений мал.

Пусть даны два вектора признаков, A и B, то косинусное сходство, $cos(\theta)$, может быть представлено используя скалярное произведение и норму:

$$sim(A,B)_{\text{Отиаи}} = \cos(\theta) = \frac{A \times B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

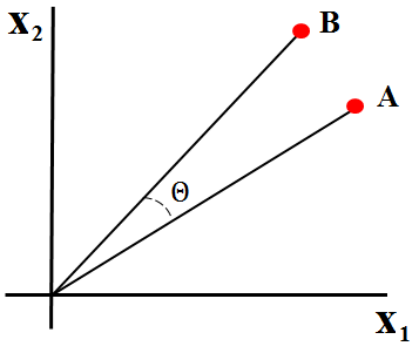

Рис. 5. Графическое представление

Косинусный коэффициент изменяется $-1 \leq \cos(\theta) \leq 1$. Если $\cos(\theta) = 1$ $\angle\theta = 0$, то вкусы пользователей похожи. Если $\cos(\theta) = -1$ $\angle\theta = 180$, то вкусы пользователей противоположны. . Если $\cos(\theta) = 0$ $\angle\theta = 90$ – зависимость между предпочтениями пользователей не просматривается.

В случае информационного поиска, косинусное сходство двух документов изменяется в диапазоне от 0 до 1, поскольку частота терма (частота появления одинаковых слов) не может быть отрицательной. Угол между двумя векторами частоты терма не может быть больше, чем 90°.

Рассмотрим пример. Пусть имеется два документа (Д1, Д2), представленные частотными векторами термов (В1 – В10) (табл. 3).

Таблица 3

	В1	В2	В3	В4	В5	В6	В7	В8	В9	В10
Д1	5	0	3	0	2	0	0	2	0	0
Д2	3	0	2	0	1	1	0	1	0	1

Вычислим косинусное подобие.

$$(Д1, Д2) = 5 \times 3 + 0 + 3 \times 2 + 0 + 2 \times 1 + 0 + 0 + 2 \times 1 + 0 + 0 = 25$$

$$\|Д1\| = \sqrt{5^2 + 0^2 + 3^2 + 0^2 + 2^2 + 0^2 + 0^2 + 2^2 + 0^2 + 0^2} = 6{,}48$$

$$\|Д2\| = \sqrt{3^2 + 0^2 + 2^2 + 0^2 + 1^2 + 1^2 + 0^2 + 1^2 + 0^2 + 1^2} = 4{,}12$$

$$sim(Д1, Д2)_{Отиаи} = \frac{25}{6{,}48 \times 4{,}12} = 0{,}94$$

Таким образом, рассматриваемые два документа очень близки с точки зрения косинусного расстояния.

Одна из причин популярности косинуснуго сходства состоит в том, что оно эффективно в качестве оценочной меры, особенно для разреженных векторов, так как необходимо учитывать только ненулевые измерения.

Коэффициент корреляции Пирсона.

Похожесть объектов i и t определяется с помощью корреляции Пирсона по формуле:

$$sim(i,t)_{\text{Пирсона}} = \frac{\sum_{u\in U}(r_{u,i} - \overline{r}_u)(r_{u,t} - \overline{r}_u)}{\sqrt{\sum_{u\in U}(r_{u,i} - \overline{r}_u)^2}\sqrt{\sum_{u\in U}(r_{u,t} - \overline{r}_u)^2}}$$

где U – множество пользователей, которые оценили объекты i и t,

$r_{u,i}$ – оценка, поставленная пользователем u объекту i,

$r_{u,t}$ – оценка, поставленная пользователем u объекту t,

$\overline{r}_u$ – средняя оценка пользователя u.

Согласно формуле, определим свойства коэффициента корреляции $sim(i,t)_{\text{Пирсона}}$:

- Коэффициент корреляции Пирсона $sim(i,t)_{\text{Пирсона}}$ изменяется в интервале от -1 до $+1$.
- Коэффициент корреляции $sim(i,t)_{\text{Пирсона}}$ безразмерен, т. е. не имеет единиц измерения.
- Величина $sim(i,t)_{\text{Пирсона}}$ указывает, как близко расположены точки к прямой линии. В частности, если $sim(i,t)_{\text{Пирсона}} = +1$ или $sim(i,t)_{\text{Пирсона}} = -1$, то имеется абсолютная (функциональная) корреляция по всем точкам, лежащим на линии (практически это маловероятно); если $sim(i,t)_{\text{Пирсона}} \cong 0$, то линейной корреляции нет (хотя может быть нелинейное соотношение). Чем ближе r к крайним точкам (± 1), тем больше степень линейной связи. При этом. если -1, то вкусы пользователей диаметрально противоположны.

Таким образом для вычисления коэффициента Пирсона нужно:

1. Рассчитать средние оценки.
2. Вычислить для каждого пользователя отклонения от среднего арифметического.
3. Поместить результаты в формулу.

Рассмотрим пример. В табл. 4 показаны оценки пользователя 1 и 2 для шести объектов.

Таблица 4

	О1	О2	О3	О4	О5	О6
П1	4	5	4	4	3	3
П2	3	3	3	2	4	5

1. Расчет средней оценки.

Необходимо рассчитать среднюю оценку для каждого пользователя. Для этого надо сложить все оценки и разделить их на количество оценок.

$$\text{Пользователь 1: } (4+5+4+4+3+3)/6 = 3{,}83;$$

$$\text{Пользователь 2: } (3+3+3+2+4+5)/6 = 3{,}33;$$

2. Вычисление отклонения от среднего арифметического.

Нужно вычесть среднюю оценку каждого пользователя из их оценок $r_{1,i} - \overline{r_1}$ и $r_{2,t} - \overline{r_2}$. На основе расчета составим таблицу 5.

Таблица 5

П1	0,17	1,17	0,17	0,17	-0,83	-0,83
П2	-0,33	-0,33	-0,33	-1,33	0,67	1,67

3. Сводим результаты в формулу.

Пусть $nr_{1,i} = r_{1,i} - \overline{r_1}$ и $nr_{2,t} = r_{2,t} - \overline{r_2}$ преобразует корреляцию Пирсона в следующий вид:

$$sim(i,t)_{\text{Пирсона}} = \frac{\sum_{u \in U}(nr_{1,i})(nr_{2,t})}{\sqrt{\sum_{u \in U}(nr_{1,i})^2}\sqrt{\sum_{u \in U}(nr_{2,t})^2}}$$

Подставив оценки получим следующее:

$$\frac{(0{,}17)(-0{,}33) + (1{,}17)(-0{,}33) + (0{,}17)(-0{,}33) + (0{,}17)(-1{,}33) + (-0{,}83)(0{,}67) + (-0{,}83)(1{,}67)}{\sqrt{(0{,}17)^2 + (1{,}17)^2 + (0{,}17)^2 + (0{,}17)^2 + (-0{,}83)^2 + (-0{,}83)^2}\sqrt{(-0{,}33)^2 + (-0{,}33)^2 + (-0{,}33)^2 + (-1{,}33)^2 + (0{,}67)^2 + (1{,}67)^2}}$$

Теперь осталось только посчитать.

$$sim(i,t)_{\text{Пирсона}} = \frac{-3{,}17}{\sqrt{2{,}83}\sqrt{3{,}56}} = -0{,}99$$

Результат показывает, что вкусы пользователя 1 и 2 расходятся. Даже можно сказать, что они противоположны.

Рассмотрим слабые стороны коэффициента корреляции Пирсона.

Расчет коэффициента корреляции Пирсона. может ввести в заблуждение, если:

- соотношение между двумя переменными нелинейное. Например, квадратичное;
- данные включают более одного наблюдения по каждому случаю;
- есть аномальные значения (выбросы);
- данные содержат ярко выраженные подгруппы наблюдений.