# Рекомендательные системы

*В статье предпринята попытка описать и классифицировать наиболее распространенные алгоритмы построения рекомендательных систем.*

Рекомендательные системы (РС) применяются в основном для предложения клиенту в реальном времени продуктов (фильмов, книг, одежды), которые, вероятно, его заинтересуют. Особенно широко РС используются в электронной коммерции. Применение РС-систем распространяется в последнее время на стационарную розничную торговлю, справочные центры, поиск по программному обеспечению, научным статьям и т.п. Это применение характеризуется предоставлением рекомендаций пользователям автоматически, на основании уже совершенных действий (покупок, выставленных рейтингов, посещений и т.д.) и приемом от них обратной связи (заказы в магазинах, переход по ссылкам и т.п.). Рекомендательные системы являются одним из важных разделов интеллектуального анализа данных – Data Mining.

**Примеры**

- **Amazon.com.** Если ресурс amazon.com не может идентифицировать или еще «не знает» пользователя, то этот пользователь получит самые общие рекомендации. В процессе совершения покупок информация о пользователе накапливается, что улучшает качество рекомендаций с учетом индивидуальных характеристик клиента. Используется алгоритм совместной фильтрации, основанной на товарах, матрица показателей сходства товаров (косинусы углов между векторами товаров (14)) формируется в отложенном режиме.
- **Google PageRank.** Здесь «похожие» страницы связаны друг с другом, возможно, опосредованной системой ссылок. Первоначальный алгоритм PageRank не принимал во внимание предпочтения пользователей, но его расширения и альтернативы могут использовать историю поисковых запросов и пользовательскую навигацию для улучшения рекомендаций.

Другие примеры: Музыка на Yahoo!, Cinemax.com, Moviecritic, TV Recommender, Video Guide, CDnow.com и проч.

## Сбор данных

В рекомендательных системах используется явный или неявный сбор данных. При явном сборе от пользователя требуется заполнять опросные анкеты для выявления предпочтений, а при неявном сборе для выявления предпочтений пользователя и составления рейтингов происходит автоматическое протоколирование его действий. Самый очевидный способ неявного сбора информации характерен для систем электронной коммерции, где рейтинг товара у пользователя оценивается в зависимости от количества заказанных единиц, включенных пользователем в свой заказ.

## Формирование рекомендаций

Различают следующие подходы к формированию рекомендаций:

- На основании содержания – рекомендации формируются для товаров, похожих на товары, уже заказанные клиентом, или на товары, заказываемые похожими клиентами. Степень

похожести оценивается на основании характеристик товаров и клиентов. Для товаров это могут быть: сюжет, режиссер, киношкола (для фильмов); общее музыкальное направление, стиль (для музыки); функциональное назначение, категория, ценовая группа (для товаров). Для клиентов характеристиками, определяющими их похожесть, могут быть: демографические данные, предпочтения из заполненных анкет и т.д. В этом подходе используются методы кластеризации товаров или клиентов, формирование между ними связей и связанных структур, а также классификационные алгоритмы Data Mining.

- На основе транзакций – рекомендации формируются на основании пользовательского поведения, т.е. товары считаются похожими, если часто входят вместе в одну транзакцию, а клиенты считаются похожими, если совершают схожие покупки. Системы выработки рекомендаций на основании транзакций называют системами совместной фильтрации (CF – collaborative filtering).

Иногда также используется комбинированный подход ([4], [6]). Например, неизвестные рейтинги исследуемого пользователя вычисляются методом, основанном на транзакциях, при этом все непроставленные другими пользователями рейтинги учитываются в алгоритме с применением модели на основании содержания.

Большинство научных публикаций, связанных с алгоритмами рекомендательных систем, основаны на совместной фильтрации.

Также при формировании рекомендаций используются два следующих подхода:

- Основанный на всех данных (Memory-based) – рекомендации формируются на основании вычисления некоторой меры по всем накопленным данным. Этот подход проще, показал высокую точность на практике и обладает преимуществом инкрементального учета новых данных (новые транзакции просто добавляются в базу данных и учитываются при формировании прогноза наряду с имеющимися). Однако, этот подход сложен для вычисления с точки зрения времени и ресурсов памяти. Также этот подход не может предоставить описательный анализ существующих закономерностей, дать б**о**льшее понимание имеющихся данных и объяснить прогноз.
- Основанный на моделях (Model-based) – сначала формируется описательная модель предпочтений пользователей, товаров и взаимосвязи между ними, а затем формируются рекомендации на основании полученной модели. Преимуществом такого подхода является наличие модели, дающей б**о**льшее понимание формируемых рекомендаций и наличия взаимосвязей в данных, а также тот факт, что 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. Однако, такие модели не поддерживают инкрементального обучения (появление новых данных требует пересчета всей модели) и в основном показывают меньшую точность прогноза, чем Memory-based.

# Совместная фильтрация

## Алгоритмы, основанные на данных

Идея совместной фильтрации (CF) состоит в предположении, что похожие клиенты совершают схожие покупки, а похожие товары покупаются клиентами совместно.

В этой связи различают два подхода к совместной фильтрации:

- Фильтрация по пользователям (User-centric). В этом случае неизвестный рейтинг товару выставляется на основании рейтингов, которые были проставлены тому же товару пользователями, похожими на данного. Этот подход реализуется в два шага:
  1. Найти пользователей, которые совершили аналогичные покупки, как и данный пользователь.
  2. Предложить товары с максимальным рейтингом среди товаров, выбираемых похожими пользователями.
- Фильтрация по товарам (Item-centric). Неизвестный рейтинг товару выставляется на основании рейтингов других похожих товаров, уже включенных пользователем в заказ. Этот подход реализуется также в два шага:
  1. Построить матрицу товаров для определения степени схожести между товарами.
  2. Используя степень схожести предложить товары, похожие на уже заказанные данным пользователем.
- Используется гибридный подход.

Далее мы будем использовать следующие обозначения.
Пусть у нас имеются данные об m товарах и n транзакциях. Обозначим каждую i-ую транзакцию m-мерным вектором $X_i := \left(x_{i,1},...,x_{i,m}\right)$, где $x_{i,j}, i \in \{1,...,n\}; j \in \{1,...,m\}$-рейтинг j-того товара, приобретенного в i-ой транзакции, а каждый j-ый товар n-мерным вектором $Y_j := \left(x_{1,j},...,x_{n,j}\right)$ рейтингов j-ого товара во всех транзакциях.

### *Фильтрация по транзакциям*

Если мы рассматриваем набор транзакций $X_i, i \in \{1,...,n\}$ как множество одинаково распределенных независимых случайных величин, то лучшим (с точки зрения минимизации среднеквадратичной ошибки) прогнозом заказа j-ого товара на u-ой транзакции, u>n в виде константы будет значение математического ожидания соответствующей случайной величины, т.е. $EX_{u,j} = EX_{1,j}$, а оценкой этого прогноза – среднее арифметическое по выборке, т.е.

$$\hat{x}_{u,j} = \frac{\sum_{i=1}^{n} x_{i,j}}{n}. \qquad \textbf{(1)}$$

Число единиц заказанного j-ого товара входит в эту сумму с одинаковым весом для каждой i-ой транзакции, что соответствует нашему предположению о равноценности каждой транзакции.
Метод фильтрации по пользователям заключается в том, что мы постулируем, что более похожие транзакции надо учитывать при прогнозировании с б**о**льшим весом в сумме, чем менее похожие. Т.е., суммируя заказы j-ого товара по всем транзакциям, мы должны учитывать вес $s\left(X,Y\right)$,

описывающий степень похожести между двумя транзакциями X и Y. Вместо формулы (1) при прогнозировании размера заказа j-товара в u-ой (новой) транзакции мы используем формулу:

$$\hat{x}_{u,j} = \frac{\sum_{i=1}^{n} s_{trans}\left(X_i, X_u\right) x_{i,j}}{\sum_{i=1}^{n} \left| s_{trans}\left(X_i, X_u\right) \right|} \qquad \textbf{(2)}$$

Чем больше сходство между транзакциями $X_i$ и $X_u$, тем с б**о**льшим весом входит число заказанного на i-ой транзакции j-ого товара во взвешенную сумму при прогнозе.

Далее мы рассмотрим версии этого алгоритма в зависимости от способа вычисления функции близости между транзакциями $s_{trans}\left(X,Y\right)$.

### К ближайших соседа

Суть метода состоит в определении К ближайших к данной транзакции транзакций и использовании среднего значения их рейтингов для прогнозирования неизвестных рейтингов в данной транзакции. Мерой близости между транзакциями служит какая-либо метрика на пространстве m-мерных векторов рейтингов товаров, например, индуцированная евклидовой нормой.

$X_u$ - исследуемая транзакция, $X_i, i \in \{1,...,n\}$ - сохраненная история завершенных транзакций.

Будем обозначать матрицей $X \in Mat\left(n,m\right)$ со строками в виде транзакций и столбцами $Y_j, j \in \{1,...,m\}$ в виде рейтингов товаров. Обозначим $d\left(X_u, X_k\right) := \sqrt{\sum_{l=1}^{m} \left(x_{u,l} - x_{k,l}\right)^2}$ - расстояние между транзакциями. Прогноз j-ого рейтинга в u-ой транзакции вычисляется как среднее значение j-ого рейтинга в К ближайших к $x_u$ транзакциях $\hat{x}_{u,j} := \frac{\sum_{i \in I_K(x_u)} x_{i,j}}{K}$, где $I_K\left(X_u\right) \subset \{1,....,n\}$, состоит из К ближайших к $x_u$ транзакций. Или, используя формулу (2), мы можем записать,

$\hat{x}_{u,j} = \frac{\sum_{i=1}^{n} s_{trans}\left(X_i, X_u\right) x_{i,j}}{\sum_{i=1}^{n} \left| s_{trans}\left(X_i, X_u\right) \right|}$, где функция близости определяется как

$$s_{trans}\left(X_u, X_i\right) := \begin{cases} 1, i \in I_K\left(X_u\right) \\ 0, i \notin I_K\left(X_u\right) \end{cases} \qquad \textbf{(3)}$$

### Взвешенные ближайшие соседи

Суть метода – использовать для усреднения не только К, а все транзакции, взятые с весами, отражающими степень близости к исследуемой транзакции.

Как и в прошлом методе $X_u$ - исследуемая транзакция, $X_i, i \in \{1,...,n\}$ - сохраненная история завершенных транзакций. $d\left(X_u, X_k\right) := \sqrt{\sum_{l=1}^{m} \left(x_{u,l} - x_{k,l}\right)^2}$ - расстояние между транзакциями.

Определим веса, отражающие степень близости между транзакциями как величины обратные расстоянию, т.е. $s_{trans}\left(X_u, X_k\right) := \frac{1}{d\left(X_u, X_k\right) + \varepsilon}$, **(4)**

где $\varepsilon > 0$ малое число, чтобы гарантировать, что знаменатель нигде не обращается в 0. Прогноз j-ого рейтинга в u-ой транзакции вычисляется как средневзвешенное значение j-ого рейтинга по всем транзакциям с весами обратно пропорциональными расстоянию между транзакциями.

$$\hat{x}_{u,j} := \frac{\sum_{i=1}^{n} s_{trans}\left(X_u, X_i\right) x_{i,j}}{\sum_{i=1}^{n} \left| s_{trans}\left(X_u, X_i\right) \right|}$$

## Метрики, основанные на углах между векторами

В качестве степени схожести между транзакциями используется корреляционный коэффициент Пирсона между m-мерными векторами транзакций (показатель линейной зависимости между центрированными векторами транзакций)

$$s_{trans}\left(X_u, X_i\right) := \frac{\sum_{k=1}^{m} \left(x_{u,k} - \overline{x}_u\right)\left(x_{i,k} - \overline{x}_i\right)}{\sqrt{\sum_{k=1}^{m} \left(x_{u,k} - \overline{x}_u\right)^2 \sum_{i=1}^{m} \left(x_{i,k} - \overline{x}_i\right)^2}}, \quad \textbf{(5)}$$

где $\overline{x}_i = \frac{\sum_{k=1}^{m} x_{i,k}}{m}$ -среднее значение рейтинга по транзакции.

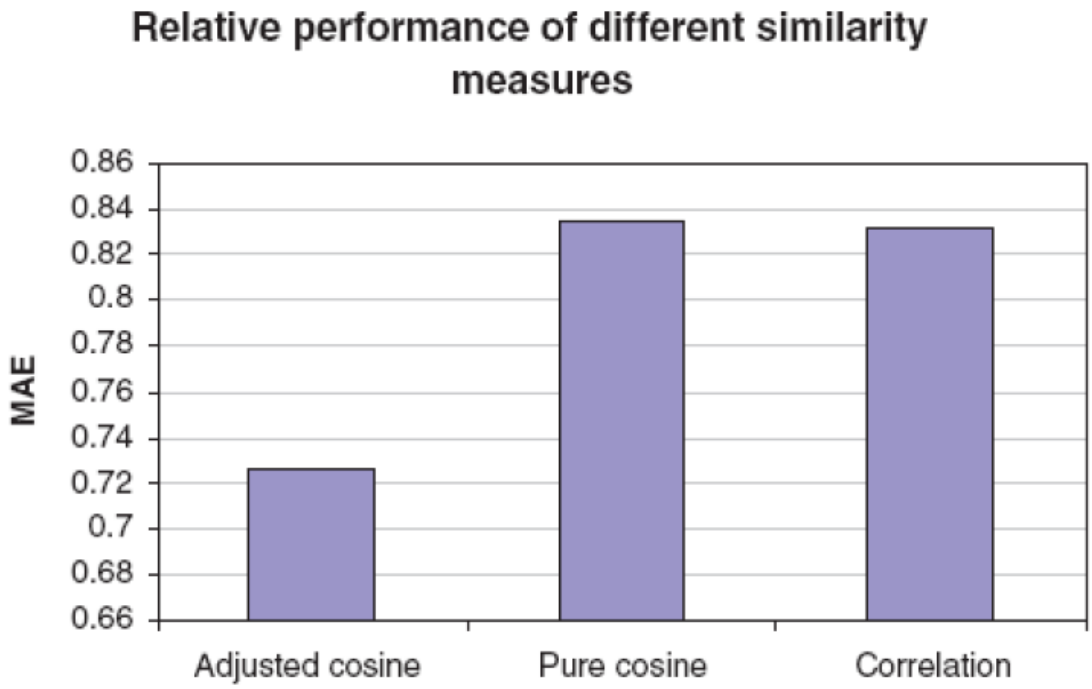
Используется также косинус между m-мерными векторами транзакций (показатель линейной зависимости между векторами транзакций)

$$s_{trans}\left(X_u, X_i\right) := \frac{\sum_{k=1}^{m} x_{u,k} x_{i,k}}{\sqrt{\sum_{k=1}^{m} x_{u,k}{}^2 \sum_{k=1}^{m} x_{i,k}{}^2}} \quad \textbf{(6)}$$

В статье [18] описано, что лучшие с точки зрения минимизации средней абсолютной ошибки прогноза результаты получаются при использовании «нормированного косинуса» в качестве меры близости:

$$s_{trans}\left(X_u, X_i\right) := \frac{\sum_{k=1}^{m} \left(x_{u,k} - \overline{y}_k\right)\left(x_{i,k} - \overline{y}_k\right)}{\sqrt{\sum_{k=1}^{m} \left(x_{u,k} - \overline{y}_k\right)^2 \sum_{i=1}^{m} \left(x_{i,k} - \overline{y}_k\right)^2}}, \quad \textbf{(7)}$$

где $\overline{y}_k := \frac{\sum_{p=1}^{n} x_{p,k}}{n}$ - средний рейтинг по k-ому столбцу. В отличие от коэффициента Пирсона (5), где значения рейтинга нормируются средним значением по строке, при использовании нормированного косинуса, значения рейтингов нормируются средними значениями по столбцам.




Практическое использование показало, что лучшие результаты получаются в том случае, если при прогнозировании рейтинга вместо его значения использовать его отклонение от среднего значения по транзакции, т.е.

$$\hat{x}_{u,j} = \overline{x}_u + \frac{\sum_{i=1}^{n} s_{trans}(X_i, X_u)\left(x_{i,j} - \overline{x}_i\right)}{\sum_{i=1}^{n} \left| s_{trans}(X_i, X_u) \right|} \quad \textbf{(8)}$$

Используя (8) мы фактически прогнозируем отклонение рейтинга j-ого товара от среднего рейтинга по транзакции.

## Сложность и ресурсоемкость

Фильтрация по пользователям показывает довольно высокую степень точности в практических применениях. Однако, недостатком всех вариантов приведенного алгоритма является его ресурсоемкость (требование к памяти) и сложность (количество вычислений, требуемое для получения рекомендаций). А именно:

- Если хранить в памяти (для быстрого доступа) векторы рейтингов для всех транзакций, т.е. матрицу n строк на m столбцов, то для среднего интернет-магазина (~1 млн. транзакций и ~ 10 тыс. товаров) потребуется хранить в памяти ~10 млрд. действительных чисел (по 8 байт), что представляется невозможным для имеющихся в распоряжении у таких магазинов компьютеров. Осуществлять доступ к этим данным с диска (БД), естественно, возможно, но это сильно замедляет выполнение операций и, соответственно, повышает требования к аппаратному обеспечению с точки зрения скорости доступа к дисковому вводу/выводу и процессору с точки зрения скорости выполнения арифметических операций.
- Каждое сравнение данной транзакции с одной из n остальных занимает порядка O(m) операций, т.е. всего надо произвести n * O(m) операций для определения степени похожести между данной транзакцией и остальными.

- Вычисление рейтинга для каждого из m товаров потребует выполнения порядка O(n) операций усреднения по транзакциям, т.е. m * O(n) операций для всех товаров.
- Итого, нам потребуется выполнить $\mathrm{m}\bullet\mathrm{O}(\mathrm{n})+\mathrm{n}\bullet\mathrm{O}(\mathrm{m})=\mathrm{O}(\mathrm{mn})$ арифметических операций для получения рейтингов всех товаров для данной транзакции, чтобы получить рекомендации.

Вышеприведенные рассуждения показывают, что фильтрация по пользователям может применяться только к относительно небольшим базам данных.

### *Фильтрация по товарам*

Идея в фильтрации по товарам состоит в выставлении неизвестного рейтинга товару в анализируемой транзакции на основании взвешенных рейтингов других товаров, входящих в эту транзакцию. Рейтинг товара получается тем больше, чем больше рейтинг у других товаров в анализируемой транзакции, которые обычно покупаются с ним совместно. Т.е. если мы пытаемся проставить рейтинг товару А и нам уже известен рейтинг товара Б в этой транзакции, то рейтинг Б будет учитываться в вычислении рейтинга А с учетом того как часто А и Б входят в одну транзакцию. Мы получаем формулу, аналогичную (1):

$$\hat{x}_{u,j}=\frac{\sum_{i=1}^{m}s_{items}\left(Y_j,Y_i\right)x_{u,i}}{\sum_{i=1}^{m}\left|s_{items}\left(Y_j,Y_i\right)\right|} \qquad \textbf{(9)}$$

Часто вместо абсолютной величины прогноза рейтинга товара прогнозируют отклонение рейтинга от средней по всем транзакциям для данного товара величину:

$$\hat{x}_{u,j}=\overline{y}_j+\frac{\sum_{i=1}^{m}s_{items}\left(Y_j,Y_i\right)\left(x_{u,i}-\overline{y}_i\right)}{\sum_{i=1}^{m}\left|s_{items}\left(Y_j,Y_i\right)\right|}\text{, где} \qquad \textbf{(10)}$$

$\overline{y}_i=\frac{\sum_{k=1}^{n}x_{k,i}}{n}$ - средний по всем транзакциям рейтинг i-ого товара.

Далее, аналогично фильтрации по пользователям, мы рассмотрим версии этого алгоритма в зависимости от способа вычисления функции близости между векторами рейтингов товаров $Y_j$ и $Y_i$.

### К ближайших соседа

Для j-ого n-мерного вектора рейтингов товара $Y_j$ вычисляются K ближайших в евклидовой норме вектора $\left\{Y_{j_1},...,Y_{j_K}\right\}$ и определяется множество $I_K\left(Y_j\right)=\left\{j_1,...,j_k\right\}$, состоящее из индексов этих K ближайших к $Y_j$ соседей. Тогда степень близости между векторами товаров $Y_j$ и $Y_i$ в (9) и (10) определяется как

$$s_{items}\left(Y_i,Y_j\right):=\begin{cases}1, i\in I_K\left(Y_j\right)\\0, i\notin I_K\left(Y_j\right)\end{cases} \qquad \textbf{(11)}$$

Т.е. усреднение в формуле (9) и (10) происходит только по K ближайшим товарам:

$$\hat{x}_{u,j} = \frac{\sum_{i \in I_K(Y_j)} x_{u,i}}{K}$$

### Взвешенные ближайшие соседи

Степень близости между векторами товаров $Y_j$ и $Y_i$ в (9) и (10) определяется как величина, обратная евклидовому расстоянию между ними, т.е.

$$s_{items}\left(Y_i, Y_j\right) := \frac{1}{d\left(Y_i, Y_j\right) + \varepsilon} \quad \textbf{(12)}$$

### Частота попарного вхождения

Весовые коэффициенты, учитывающие «близость» между n-мерными векторами рейтингов товаров $Y_j$ и $Y_i$ вычисляются как относительные частоты совместного вхождения двух товаров в одну транзакцию:

$$s_{items}\left(Y_i, Y_j\right) = \frac{\sum_{k=1}^{n} \delta\left(x_{k,i}, x_{k,j}\right)}{n}, \text{ где} \quad \textbf{(13)}$$

$\delta\left(x_{k,i}, x_{k,j}\right) = \begin{cases} 1, x_{k,i} x_{k,j} > 0 \\ 0, x_{k,i} x_{k,j} = 0 \end{cases}$ - флаг того, что i-ый и j-ый товар совместно входят в k-ую транзакцию.

### Метрики на основе углов между векторами товаров

В качестве меры близости векторов товаров используется также корреляция Пирсона, косинус или нормированный косинус угла между ними:

$$s_{items}\left(Y_i, Y_j\right) := \frac{\sum_{k=1}^{n}\left(x_{k,i} - \overline{y}_i\right)\left(x_{k,j} - \overline{y}_j\right)}{\sqrt{\sum_{k=1}^{n}\left(x_{k,i} - \overline{y}_i\right)^2 \sum_{i=1}^{n}\left(x_{k,j} - \overline{y}_j\right)^2}}, \quad \textbf{(14)}$$

$$s_{items}\left(Y_i, Y_j\right) := \frac{\sum_{k=1}^{n} x_{k,i} x_{k,j}}{\sqrt{\sum_{k=1}^{n} x_{k,i}{}^2 \sum_{i=1}^{n} x_{k,j}{}^2}}, \quad \textbf{(15)}$$

$$s_{items}\left(Y_i, Y_j\right) := \frac{\sum_{k=1}^{n}\left(x_{k,i} - \overline{x}_k\right)\left(x_{k,j} - \overline{x}_k\right)}{\sqrt{\sum_{k=1}^{n}\left(x_{k,i} - \overline{x}_k\right)^2 \sum_{i=1}^{n}\left(x_{k,j} - \overline{x}_k\right)^2}} \quad \textbf{(16)}$$

где $\overline{y}_i = \frac{\sum_{k=1}^{n} x_{k,i}}{n}$ - среднее значение рейтинга по i-ому товару, а $\overline{x}_k = \frac{\sum_{p=1}^{m} x_{k,p}}{m}$ - среднее значение рейтинга по k-ой транзакции.

## Сложность и ресурсоемкость

В отличие от алгоритма фильтрации по транзакциям, где вычисление степени близости анализируемой транзакции ко всем остальным транзакциям может производиться только в реальном времени, так как данные о текущей транзакции становятся доступными только в момент выработки рекомендаций, в алгоритме фильтрации по товарам степень близости анализируемого товара ко всем остальным товарам может быть вычислена в отложенном режиме по расписанию, так как вектора рейтингов всех товаров доступны до момента формирования рекомендации. Таким образом, разделив процесс выработки рекомендаций на отложенную стадию (вычисление степени близости товаров друг к другу) и стадию в реальном времени (вычисление рейтингов товаров), мы получим, что сложность алгоритма фильтрации по товарам на стадии формирования рекомендаций равна $O(m^2)$, что в отличие от сложности фильтрации по транзакциям O(mn) не зависит от числа транзакций. Таким образом, если число транзакций значительно превышает число товаров, то алгоритм фильтрации по товарам оказывается более эффективным с точки зрения времени формирования рекомендаций, чем алгоритм фильтрации по транзакциям благодаря возможности проведения отложенной предобработки данных.

### *Комбинированная фильтрация*

В рассмотренных выше алгоритмах фильтрации по пользователям (взвешенное усреднение рейтингов по исследуемому товару по похожим транзакциям) и фильтрации по товарам (взвешенное усреднение рейтингов похожих товаров в исследуемой транзакции) используется только одна часть информации из накопленных данных для прогнозирования неизвестных рейтингов (используется или корреляция между векторами транзакций или корреляция между векторам рейтингов товаров). В этой связи интуитивно представляется желательным объединить рейтинги как от похожих транзакций, так и от похожих товаров для более эффективного использования имеющейся информации. Также рассмотренные выше методы совместной фильтрации игнорируют информацию, которую можно получить от рейтингов товаров, похожих на исследуемый товар из других транзакций, похожих на исследуемую транзакцию. Отказ от использования этих рейтингов уменьшает возможность прогнозирования из-за недостатка похожих рейтингов. Таким образом, идея комбинированной фильтрации заключается в получении оценки неизвестного рейтинга как взвешенной суммы оценок на основании фильтрации по транзакциям, фильтрации по товарам и смешенной фильтрации (на основании рейтингов похожих товаров в похожих транзакциях).

Для получения рейтинга при смешенной фильтрации, т.е. на основании похожих товаров в похожих транзакциях, нам необходимо определить совместную степень сходства между парами транзакция - товар. Для этой цели используется величина, учитывающая степень сходства отдельно соответствующих транзакций и отдельно - товаров. Комбинированная степень сходства не превышает отдельных степеней сходства между транзакциями и товарами.

$$s_{trans-items}\left(x_{u,k},x_{i,j}\right):=\frac{1}{\sqrt{\left(1/s_{trans}\left(X_u,X_i\right)\right)^2+\left(1/s_{items}\left(Y_k,Y_j\right)\right)^2}}, \quad \textbf{(17)}$$

где $s_{trans}$ вычисляется согласно (5), (6) или (7), а $s_{items}$ – согласно (14) , (15) или (16).

Введем следующие обозначения: под множеством $S_{tran}\left(X\right)\subset\{1,...,n\}$ будем обозначать индексы K самых близких с точки зрения (5), (6) или (7) к X транзакций из $X_1,...,X_n$. Под $S_{item}\left(Y\right)\subset\{1,...,m\}$ будем обозначать индексы K самых близких с точки зрения (14), (15) или (16) к Y товара из $Y_1,...,Y_m$. Сортируя и переиндексируя строки и колонки матрицы рейтингов по степени схожести к исследуемой транзакции и к исследуемому товару и оставив в ней для последующего анализа не более K строк и столбцов, мы получим матрицу похожих транзакций и товаров $M_{u,k}:=\left\{x_{i,j}\middle|i\in S_{tran}\left(X_u\right),j\in S_{item}\left(Y_k\right)\right\}$.

Степень сходства по товарам

Степень сходства по транзакциям

| Неизвестный рейтинг | Фильтрация по товарам | Фильтрация по товарам | Фильтрация по товарам | Фильтрация по товарам |
|---|---|---|---|---|
| Фильтрация по транзакциям | Смешанная фильтрация | Смешанная фильтрация | Смешанная фильтрация | Смешанная фильтрация |
| Фильтрация по транзакциям | Смешанная фильтрация | Смешанная фильтрация | Смешанная фильтрация | Смешанная фильтрация |
| Фильтрация по транзакциям | Смешанная фильтрация | Смешанная фильтрация | Смешанная фильтрация | Смешанная фильтрация |
| Фильтрация по транзакциям | Смешанная фильтрация | Смешанная фильтрация | Смешанная фильтрация | Смешанная фильтрация |

Если для оценки неизвестного рейтинга мы будем учитывать только самый первый столбец с использованием меры близости по транзакциям, то получим оценку по транзакциям:

$$\hat{x}_{u,k}^{\ trans}=\frac{\sum_{i=1,i\neq u}^{K}s_{trans}\left(X_u,X_i\right)x_{i,k}}{\sum_{i=1,i\neq u}^{K}s_{trans}\left(X_u,X_i\right)} \quad \textbf{(18)}$$

Если для оценки неизвестного рейтинга мы будем учитывать только первую строку с использованием меры близости по товарам, то получим оценку по товарам:

$$\hat{x}_{u,k}^{\ items}=\frac{\sum_{j=1,j\neq k}^{K}s_{items}\left(Y_j,Y_k\right)x_{u,j}}{\sum_{j=1,j\neq k}^{K}s_{items}\left(Y_j,Y_k\right)} \quad \textbf{(19)}$$

Если для оценки рейтинга мы будем учитывать все элементы матрицы кроме первой строки и первого столбца, т.е. похожие товары, участвующие в похожих транзакциях, то получим оценку:

$$\hat{x}_{u,k}^{\ trans-items} = \frac{\sum_{i=1,i\neq u}^{K}\sum_{j=1,j\neq k}^{K} s_{trans-items}\left(x_{u,k},x_{i,j}\right)x_{i,j}}{\sum_{i=1,i\neq u}^{K}\sum_{j=1,j\neq k}^{K} s_{trans-items}\left(x_{u,k},x_{i,j}\right)} \quad \textbf{(20)}$$

Итоговая комбинированная оценка неизвестного рейтинга получается в виде взвешенной оценки из (18), (19) и (20):

$\hat{x}_{u,k} = \lambda(1-\delta)\hat{x}_{u,k}^{\ trans} + (1-\lambda)(1-\delta)\hat{x}_{u,k}^{\ items} + \delta\hat{x}_{u,k}^{\ trans-items}$ , где **(21)**

где $\lambda,\delta \in [0,1]$ – веса учета соответствующих оценок. При $\delta = 0, \lambda = 1$ мы получаем фильтрацию по транзакциям, при $\delta = 0, \lambda = 0$ – фильтрацию по товарам, при $\delta \neq 0$ мы также учитываем рейтинги сходных товаров из сходных транзакций.

### *Понижение размерности*

Как уже было отмечено, основным недостатком алгоритмов совместной фильтрации является необходимость выполнения большого количества операций для вычисления степени похожести товаров или транзакций и для усреднения рейтингов по товарам или по транзакциям при прогнозировании неизвестного рейтинга. Для уменьшения трудоемкости операций усреднения (т.е. в формулах (2), (9)) на практике используется не усреднение по всем транзакциям (2) и не по всем товарам (9), а лишь по K наиболее похожим. Численные эксперименты с реальными базами данных ([4]) показали, что выбор числа K наиболее похожих транзакций или товаров, по которым производится усреднение для вычисления рейтинга, оказывает сильное влияние на точность рекомендаций. Общей тенденцией является увеличение точности при начальном увеличении числа K, а затем, после достижения максимума, точность стабилизируется или плавно ухудшается. Ухудшение точности при дальнейшем увеличении K объясняется тем фактом, что все б**о**льшее количество «непохожих» транзакций или товаров принимается к рассмотрению. Оптимальное число K для фильтрации по товарам в среднем имеет значение около 10, в то время как по транзакциям – на порядок больше. Таким образом рассмотрение только K ближайших транзакций или товаров вместо всех имеющихся в распоряжении не только ускоряет процесс вычисления неизвестного рейтинга, но и увеличивает точность прогноза.

Для уменьшения сложности вычисления степени схожести векторов товаров или транзакций (например, в формулах (5) или (14)) используется подход понижения размерности матрицы транзакций-товаров, основанный на разложении этой матрицы по сингулярным значениям. Разложение по сингулярным значениям (SVD – Singular Value Decomposition) представляет собой представление матрицы $A \in Mat(n,m)$ с рангом $r := ran(A) \leq \min\{n,m\}$ в виде $A = USV^t$, где матрицы $U \in Mat(n,r)$ и $V \in Mat(m,r)$ состоят из ортонормальных столбцов, являющихся собственными векторами при ненулевых собственных значениях матриц $AA^t$ и $A^tA$ соответственно, а $S = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \begin{matrix} . & 0 \\ 0 & . \end{matrix} & 0 \\ 0 & 0 & \lambda_r \end{pmatrix} \in Mat(r,r)$-диагональная матрица с положительными диагональными элементами $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_r > 0$, отсортированными в порядке убывания. Диагональные элементы матрицы S $\lambda_1, \lambda_2, \ldots, \lambda_r$ представляют собой ненулевые собственные

значения, соответствующие собственным векторам $AA^t$ и $A^tA$ (столбцам U и V). Столбцы матрицы U представляют собой, таким образом, ортонормальный базис пространства столбцов матрицы A, а столбцы матрицы V – ортонормальный базис пространства строк матрицы A. Важным свойством SVD-разложения является тот факт, что если для d<r преобразовать матрицу S в матрицу $S_d = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \begin{matrix} . & 0 \\ 0 & . \end{matrix} & 0 \\ 0 & 0 & \lambda_d \end{pmatrix} \in Mat(d,d)$, состоящую только из d наибольших диагональных элементов, а также оставить в матрице U и V только d первых столбцов, т.е. преобразовать их в $U_d \in Mat(n,d)$ и $V_d \in Mat(m,d)$, то матрица $A_d := U_d S_d V_d^t$ будет являться лучшей аппроксимации матрицы A относительно нормы Фробениуса среди всех матриц с рангом d, т.е. $\|A - A_d\|_{Frobenius} \le \|A - A'\|_{Frobenius} \; \forall A' \in Mat(n,m), ran(A') = d$.

Суть понижения размерности при помощи SVD-разложения исходной матрицы транзакций-товаров A состоит в следующем: сначала строится разложение $A = USV^t$, затем для фиксированного выбранного d<<ran(A) мы получаем лучшую d-ранговую аппроксимацию матрицы A в форме $A \approx A_d = U_d S_d V_d^t$.

В случае фильтрации по товарам каждый j-ый столбец $Y_j$ матрицы A, соответствующий рейтингам j-ого товара, аппроксимируется j-ым столбцом матрицы $A_d$, который представляет собой проекцию вектора $Y_j$ на пространство, образованное d ортонормальными столбцами матрицы $U_d$ с коэффициентами разложения $C_j := \left(S_d V_d^t\right)_j$, соответствующими j-ому d-мерному вектору-столбцу матрицы $S_d V_d^t$. Таким образом, вместо n-мерного вектора j-ого товара $Y_j$ мы рассматриваем d-мерный вектор $C_j$, представляющий собой вектор коэффициентов разложения проекции $Y_j$ по базису $U_d$. Используя описанный подход, для определения степени похожести векторов товаров $Y_u$ и $Y_k$ мы вычисляем степень похожести их d-мерных аппроксимаций, т.е. вычисляем, например, косинусы между векторами коэффициентов $C_u$ и $C_k$ разложения исходных векторов товаров $Y_u$ и $Y_k$ по базису $U_d$ в виде, аналогичном (15):

$$s_{items}^{reduced}\left(Y_u, Y_k\right) := \frac{\sum_{i=1}^{d} c_{i,u} c_{i,k}}{\sqrt{\sum_{i=1}^{d} {c_{i,u}}^2 \sum_{i=1}^{d} {c_{i,k}}^2}} \quad \textbf{(22)}$$

В отличие от (14) нам требуется O(d) операций для вычисления степени похожести между векторами товаров вместо O(n), что значительно ускоряет вычисления при d<<n.

Неизвестный рейтинг прогнозируется по формуле, аналогичной (9), но вместо исходных рейтингов $x_{u,i}$ в ней участвуют элементы $x_{u,i}^d$ из аппроксимирующей матрицы $A_d$

$$\hat{x}_{u,j} = \frac{\sum_{i=1, i \in I_K(Y_j)}^{m} s_{items}^{reduced}\left(Y_j, Y_i\right) x_{u,i}^d}{\sum_{i=1, i \in I_K(Y_j)}^{m} \left| s_{items}^{reduced}\left(Y_j, Y_i\right) \right|} \quad \textbf{(23)}$$

Аналогично, в случае фильтрации по транзакциям, каждая i-ая строка матрицы A, соответствующая i-ой транзакции, аппроксимируется i-ой строкой матрицы $A_d$ , которая представляет собой линейную комбинацию ортонормальных строк матрицы $V_d^t$ с коэффициентами, соответствующими d-мерной i-ой строке $B_i := (U_d S_d)_i$ матрицы $U_d S_d$ . Т.е. вместо определения степени похожести m-мерных векторов транзакций $X_u$ и $X_k$ мы сравниваем d-мерные коэффициенты их разложения $B_u$ и $B_k$ по базису строк матрицы $V_d^t$ . Вместо, например, 6) мы получаем:

$$s_{trans}^{reduced}(X_u, X_k) := \frac{\sum_{j=1}^{d} b_{u,j} b_{k,j}}{\sqrt{\sum_{j=1}^{d} b_{u,j}^2 \sum_{j=1}^{d} b_{k,j}^2}} \quad \textbf{(24)}$$

В отличие от (6) нам потребуется не O(m) операций, а O(d), что значительно ускоряет вычисления при d<<m.

Неизвестный рейтинг прогнозируется по формуле, аналогичной (2), но вместо исходных рейтингов $x_{u,i}$ в ней участвуют элементы $x_{u,i}^d$ из аппроксимирующей матрица $A_d$

$$\hat{x}_{u,j} = \frac{\sum_{i=1, i \in I_K(X_u)}^{n} s_{trans}^{reduced}(X_i, X_u) x_{i,j}^d}{\sum_{i=1, i \in I_K(X_u)}^{n} \left| s_{trans}^{reduced}(X_i, X_u) \right|} \quad \textbf{(25)}$$

В [4] было показано, что ранг матрицы аппроксимации d оказывает большое значение на точность получаемого прогноза. Это число должно быть достаточно маленьким, чтобы оказать заметное воздействие на ускорение выполнения вычислений и чтобы минимизировать переобучение с одной стороны, и достаточно большим, чтобы содержать важные объективные взаимосвязи между транзакциями и товарами, содержащимися в исходных данных. Точность прогнозирования следует следующему правилу: при увеличении числа d точность прогноза растет и быстро достигает своего максимума (в среднем около d=6), а затем точность ухудшается. Причина ухудшения точности прогноза при увеличении ранга аппроксимирующей матрицы объясняется переобучением (излишним усложнением) модели, ведущим не к выявлению объективных зависимостей между товарам и транзакциями, а к подгонке к обучающим данным.

Таким образом, использование только ограниченного числа наиболее похожих товаров и транзакций, а также аппроксимация матрицы транзакций-товаров матрицей значительно меньшего ранга не только упрощает вычисления, но и увеличивает точность прогнозирования из-за уменьшения влияния факторов переобучения модели.

### *Метод персональной диагностики*

Описание метода персональной диагностики (PD) основано на работе [5]. Суть метода заключается в том, что мы рассматриваем анализируемую транзакцию $X_u$ как будто она была выбрана случайно с равномерным распределением из генеральной совокупности $X_1, ..., X_n$

транзакций с добавлением белого гауссовского шума. Т.е. помимо наблюдаемой случайной величины $X_u$, соответствующей рейтингам товаров в анализируемой транзакции, у нас есть еще ненаблюдаемая случайная величина $X_u^{true}$, которая описывает вариантом какой транзакции является $X_u$ «на самом деле». Величина $X_u^{true}$ может принимать значения $X_1,...,X_n$ с вероятностью $\frac{1}{n}$ (если все вектора транзакций попарно различны). Ее можно также интерпретировать как флаг принадлежности к типу пользователя или к одноэлементному кластеру, соответствующему каждой транзакции. Итак, априорная вероятность принадлежности u-ой транзакции к i-ому типу пользователя принимается равной $P\left(X_u^{true}=X_i\right)=\frac{1}{n}$ **(26)**

Условная плотность распределения рейтинга k-ого товара в u-ой транзакции при известном значении «истинного» значения рейтинга k-ого товара в u-ой транзакции (или рейтинга k-ого товара для транзакции, вариантом которой u-ая транзакция является «на самом деле») принимается нормальной с математическим ожиданием, равным рейтингу «истинной» транзакции и с дисперсией $\sigma_k$, являющейся свободным параметром:

$$P\left(x_{u,k}=x\middle|x_{u,k}^{true}=y\right)=N\left(y,\sigma_k\right)(x)=\frac{1}{\sqrt{2\pi}\sigma_k}e^{\frac{(x-y)^2}{2{\sigma_k}^2}} \quad \textbf{(27)}$$

Кроме того, при фиксированном известном значение «истинного» рейтинга k-ого товара в u-ой транзакции (рейтинга k-ого товара в транзакции, вариантом которой является u-ая трназакция), случайная величина наблюдаемого рейтинга k-ого товара в u-ой транзакции не зависит условно ни от одной случайной величины в модели, т.е.

$$P\left(x_{u,k}=x,Z\middle|x_{u,k}^{true}=y\right)=P\left(x_{u,k}=x\middle|x_{u,k}^{true}=y\right)P\left(Z\middle|x_{u,k}^{true}=y\right),\forall Z:\Omega\to\mathbb{R} \quad \textbf{(28)}$$

На основании этого условная плотность вероятности рейтинга k-ого товара в u-ой транзакции при известных значениях некоторого набора рейтингов других товаров в этой транзакции равна:

$$P\left(x_{u,k}=x\middle|x_{u,j_1}=x_1,...,x_{u,j_s}=x_s\right)=\frac{P\left(x_{u,k}=x,x_{u,j_1}=x_1,...,x_{u,j_s}=x_s\right)}{P\left(x_{u,j_1}=x_1,...,x_{u,j_s}=x_s\right)}=$$

$$=\frac{\sum_{i=1}^{n}P\left(x_{u,k}=x,x_{u,j_1}=x_1,...,x_{u,j_s}=x_s,X_u^{true}=X_i\right)}{\sum_{i=1}^{n}P\left(x_{u,j_1}=x_1,...,x_{u,j_s}=x_s,X_u^{true}=X_i\right)}=$$

$$\frac{\sum_{i=1}^{n}P\left(X_u^{true}=X_i\right)P\left(x_{u,k}=x\middle|X_u^{true}=X_i\right)P\left(x_{u,j_1}=x_1,...,x_{u,j_s}=x_s\middle|X_u^{true}=X_i,x_{u,k}=x\right)}{\sum_{i=1}^{n}P\left(X_u^{true}=X_i\right)P\left(x_{u,j_1}=x_1,...,x_{u,j_s}=x_s\middle|X_u^{true}=X_i\right)} \quad \textbf{(29)}$$

Далее получаем:

$$P\left(x_{u,j_1}=x_1,...,x_{u,j_s}=x_s\middle|X_u^{true}=X_i,x_{u,k}=x\right)=$$

$$=P\left(x_{u,j_1}=x_1\middle|X_u^{true}=X_i,x_{u,k}=x\right)\cdot...\cdot P\left(x_{u,j_s}=x_s\middle|X_u^{true}=X_i,x_{u,k}=x,x_{u,j_1}=x_1,...,x_{u,j_{s-1}}=x_{s-1}\right) \quad \textbf{(30)}$$

Из условной независимости (28) получаем с учетом (30):

$$P\left(x_{u,j_1}=x_1,...,x_{u,j_s}=x_s\left|X_u^{true}=X_i,x_{u,k}=x\right.\right)=P\left(x_{u,j_1}=x_1\left|x_{u,j_1}^{true}=x_{i,j_1}\right.\right)\cdot...\cdot P\left(x_{u,j_s}=x_s\left|x_{u,j_s}^{true}=x_{i,j_s}\right.\right)$$

Аналогично:

$$P\left(x_{u,j_1}=x_1,...,x_{u,j_s}=x_s\left|X_u^{true}=X_i\right.\right)=P\left(x_{u,j_1}=x_1\left|x_{u,j_1}^{true}=x_{i,j_1}\right.\right)\cdot...\cdot P\left(x_{u,j_s}=x_s\left|x_{u,j_s}^{true}=x_{i,j_s}\right.\right)$$

После подстановки двух последних равенств в (29) и учитывая (26), получаем:

$$P\left(x_{u,k}=x\left|x_{u,j_1}=x_1,...,x_{u,j_s}=x_s\right.\right)=$$

$$=\frac{\sum_{i=1}^{n}P\left(x_{u,k}=x\left|x_{u,k}^{true}=x_{i,k}\right.\right)P\left(x_{u,j_1}=x_1\left|x_{u,j_1}^{true}=x_{i,j_1}\right.\right)\cdot...\cdot P\left(x_{u,j_s}=x_s\left|x_{u,j_s}^{true}=x_{i,j_s}\right.\right)}{\sum_{i=1}^{n}P\left(x_{u,j_1}=x_1\left|x_{u,j_1}^{true}=x_{i,j_1}\right.\right)\cdot...\cdot P\left(x_{u,j_s}=x_s\left|x_{u,j_s}^{true}=x_{i,j_s}\right.\right)}\quad\textbf{(31)}$$

С учетом (27) получаем из (31):

$$P\left(x_{u,k}\left|x_{u,j_1},...,x_{u,j_s}\right.\right)=\frac{\left(\frac{1}{\sqrt{2\pi}}\right)^{s+1}\frac{1}{\sigma_k\sigma_{j_1}\cdot...\cdot\sigma_{j_s}}\sum_{i=1}^{n}e^{-\frac{\left(x_{u,k}-x_{i,k}\right)^2}{2\sigma_k^{\ 2}}}e^{-\sum_{p=1}^{s}\frac{\left(x_{u,j_p}-x_{i,j_p}\right)^2}{2\sigma_{j_p}^{\ 2}}}}{\left(\frac{1}{\sqrt{2\pi}}\right)^{s}\frac{1}{\sigma_{j_1}\cdot...\cdot\sigma_{j_s}}\sum_{i=1}^{n}e^{-\sum_{p=1}^{s}\frac{\left(x_{u,j_p}-x_{i,j_p}\right)^2}{2\sigma_{j_p}^{\ 2}}}}=$$

$$=\frac{\sum_{i=1}^{n}\frac{1}{\sqrt{2\pi}\sigma_k}e^{-\frac{\left(x_{u,k}-x_{i,k}\right)^2}{2\sigma_k^{\ 2}}}e^{-\sum_{p=1}^{s}\frac{\left(x_{u,j_p}-x_{i,j_p}\right)^2}{2\sigma_{j_p}^{\ 2}}}}{\sum_{i=1}^{n}e^{-\sum_{p=1}^{s}\frac{\left(x_{u,j_p}-x_{i,j_p}\right)^2}{2\sigma_{j_p}^{\ 2}}}}\quad\textbf{(32)}$$

Для краткости обозначим $M_i:=e^{-\sum_{p=1}^{s}\frac{\left(x_{u,j_p}-x_{i,j_p}\right)^2}{2\sigma_{j_p}^{\ 2}}}$ **(33)**

Прогнозом неизвестного рейтинга $x_{u,k}$ мы будем считать его условное математическое ожидание с учетом известных значений рейтингов, т.е. $E\left(x_{u,k}\left|x_{u,j_1},...,x_{u,j_s}\right.\right)$, т.к. именно эта величина является лучшей аппроксимацией с точки зрения $L_2$-нормы случайной величины $x_{u,k}$ в пространстве измеримых функций от $x_{u,j_1},...,x_{u,j_s}$ (т.е. среди функций от имеющейся информации). Зная условную плотность $x_{u,k}$ из (32), мы можем вычислить прогноз.

$$\hat{x}_{u,k}:=E\left(x_{u,k}\left|x_{u,j_1},...,x_{u,j_s}\right.\right)=\int x_{u,k}P\left(x_{u,k}\left|x_{u,j_1},...,x_{u,j_s}\right.\right)dx_{u,k}=$$

$$=\frac{\sum_{i=1}^{n}M_i\frac{1}{\sqrt{2\pi}\sigma_k}\int x_{u,k}e^{-\frac{\left(x_{u,k}-x_{i,k}\right)^2}{2\sigma_k^{\ 2}}}dx_{u,k}}{\sum_{i=1}^{n}M_i}=\frac{\sum_{i=1}^{n}M_ix_{i,k}}{\sum_{i=1}^{n}M_i}\quad\textbf{(34)}$$

Т.е. $\hat{x}_{u,k}=\frac{\sum_{i=1}^{n}M_ix_{i,k}}{\sum_{i=1}^{n}M_i}$ **(35)**

Таким образом, метод персональной диагностики представляет собой разновидность метода фильтрации по транзакциям (формулы (35) и (2) идентичны), в котором в качестве меры близости между транзакциями принимается значение из формулы (33), т.е.

$$s_{trans}\left(X_i, X_u\right) = e^{-\sum_{p=1}^{s}\frac{\left(x_{u,j_p}-x_{i,j_p}\right)^2}{2\sigma_{j_p}^{\ 2}}} \quad \textbf{(36)}$$

Описанный метод интересен тем, что имеет в качестве основы ясную теоретико-вероятностную модель.

### *Нормализация исходных данных*

Часто исходная матрица рейтингов нормализуется: строки и столбцы приводятся к векторам с нулевым средним. Сначала мы нормализуем матрицу, вычитая средние рейтинг по товарам:

$norm_I\left(x_{u,k}\right) := x_{u,k} - \frac{1}{n}\sum_{i=1}^{n} x_{i,k}$ , далее мы преобразуем полученную матрицу, вычитая из нее средние значения по транзакциям:

$$norm_{I,U}\left(x_{u,k}\right) := norm_I\left(x_{u,k}\right) - \frac{1}{m}\sum_{j=1}^{m} norm_I\left(x_{u,k}\right) =$$

$$= x_{u,k} - \frac{1}{n}\sum_{i=1}^{n} x_{i,k} - \frac{1}{m}\sum_{j=1}^{m}\left(x_{u,k} - \frac{1}{n}\sum_{i=1}^{n} x_{i,k}\right) =$$

$$= x_{u,k} - \frac{1}{n}\sum_{i=1}^{n} x_{i,k} - \frac{1}{m}\sum_{j=1}^{m} x_{u,k} + \frac{1}{mn}\sum_{i=1}^{n}\sum_{i=1}^{n} x_{i,k}$$

Далее везде вместо элементов $x_{u,k}$ мы используем нормализованные элементы

$$x_{u,k} - \frac{1}{n}\sum_{i=1}^{n} x_{i,k} - \frac{1}{m}\sum_{j=1}^{m} x_{u,k} + \frac{1}{mn}\sum_{i=1}^{n}\sum_{i=1}^{n} x_{i,k}\ . \quad \textbf{(37)}$$

## Алгоритмы, основанные на моделях

В отличие от ранее рассмотренной совместной фильтрации, оперирующей всей базой данных для создания рекомендаций (Memory-based), алгоритмы, основанные на моделях (Model-based) используют базу данных для обучения или настройки модели, которая затем используется при составлении рекомендаций. В процессе обучения модели вычисляется классификационная функция, которая в зависимости от имеющихся в распоряжении рейтингов позволяет получить значения неизвестных рейтингов. Вид классификационной функции (модель) обновляется регулярно по расписанию для учета всех новых поступивших транзакций, например, в периоды наименьшей загрузки вычислительных ресурсов. Разбиения процесса на отложенное периодическое обучение (ресурсоемкое вычисление и сохранение значений параметров и структуры модели) и прогнозирование рейтингов в реальном времени (относительно незатратное) позволяет оптимизировать время выполнения операций. К тому же полученная модель данных имеет самостоятельную ценность кроме возможности прогнозирования, так как позволяет получить описательный анализ имеющихся данных и зависимостей в них.

### *Упрощенный алгоритм Байеса (Naïve Bayes)*

Пусть значения рейтингов могут принимать одно из конечного набора значений $V=\{v_1,...,v_k\}$. Тогда условная вероятность того, что в u-ой транзакции рейтинг j-ого товара будет $v_j$ при условии, что некое подмножество других товаров имеет заданный набор значений рейтингов в данной транзакции, будет равна:

$$P\left(x_{u,j}=v_j\left|x_{u,j_1}=v_{j_1},...,x_{u,j_s}=v_{j_s}\right.\right)=\frac{P\left(x_{u,j}=v_j,x_{u,j_1}=v_{j_1},...,x_{u,j_s}=v_{j_s}\right)}{P\left(x_{u,j_1}=v_{j_1},...,x_{u,j_s}=v_{j_s}\right)}=$$
$$=\frac{P\left(x_{u,j_1}=v_{j_1},...,x_{u,j_s}=v_{j_s}\left|x_{u,j}=v_j\right.\right)P\left(x_{u,j}=v_j\right)}{P\left(x_{u,j_1}=v_{j_1},...,x_{u,j_s}=v_{j_s}\right)} \quad \textbf{(38)}$$

Оптимальная (с точки зрения минимизации вероятности принятия неправильного решения) решающая функция $\pi$ имеет вид

$$\pi\left(x_{u,j_1},...,x_{u,j_s}\right)=v_j \Leftrightarrow P\left(x_{u,j}=v_j\left|x_{u,j_1},...,x_{u,j_s}\right.\right)\geq P\left(x_{u,j}=v\left|x_{u,j_1},...,x_{u,j_s}\right.\right),\forall v\in V, \quad \textbf{(39)}$$

т.е. на основании имеющегося неполного набора $x_{u,j_1},...,x_{u,j_s}$ проставленных рейтингов мы принимаем решение, что отсутствующий рейтинг $x_{u,j}$ принимает значение $v_j$ тогда и только тогда, когда условная вероятность этого рейтинга для $x_{u,j}$ при известном частичном наборе $x_{u,j_1},...,x_{u,j_s}$ максимальна.

Упрощенность данного подхода состоит в дополнительном допущении об условной независимости двух любых рейтингов при условии третьего, т.е.:

$$P\left(x_{u,j}=v_t,x_{u,p}=v_r\left|x_{u,l}=v_q\right.\right)=P\left(x_{u,j}=v_t\left|x_{u,l}=v_q\right.\right)P\left(x_{u,p}=v_r\left|x_{u,l}=v_q\right.\right) \quad \textbf{(40)}$$

С учетом (40) мы можем переписать (38) как

$$P\left(x_{u,j}=v_j\left|x_{u,j_1}=v_{j_1},...,x_{u,j_s}=v_{j_s}\right.\right)=$$
$$=\frac{P\left(x_{u,j}=v_j\right)P\left(x_{u,j_1}=v_{j_1}\left|x_{u,j}=v_j\right.\right)\cdot...\cdot P\left(x_{u,j_s}=v_{j_s}\left|x_{u,j}=v_j\right.\right)}{P\left(x_{u,j_1}=v_{j_1},...,x_{u,j_s}=v_{j_s}\right)} \quad \textbf{(41)}$$

Знаменатель в правой части (41) не зависит от $v_j$, т.е. для нахождения максимума условной вероятности по $v_j$ в левой части (41) надо найти максимум числителя правой части по $v_j$. Т.е. мы принимаем решение, что $x_{u,j}=v_j$ при условии $x_{u,j_1}=v_{j_1},...,x_{u,j_s}=v_{j_s}$ тогда и только тогда, когда

$$P\left(x_{u,j}=v_j\right)P\left(x_{u,j_1}=v_{j_1}\left|x_{u,j}=v_j\right.\right)\cdot...\cdot P\left(x_{u,j_s}=v_{j_s}\left|x_{u,j}=v_j\right.\right)\geq$$
$$\geq P\left(x_{u,j}=v\right)P\left(x_{u,j_1}=v_{j_1}\left|x_{u,j}=v\right.\right)\cdot...\cdot P\left(x_{u,j_s}=v_{j_s}\left|x_{u,j}=v\right.\right),\forall v\in V \quad \textbf{(42)}$$

Оценка максимального правдоподобия для вероятностей и условных вероятностей в (42) соответствует частотам соответствующих событий в данных. Таким образом

$$P\left(x_{u,j}=v_j\right)\approx\frac{\sum_{i=1}^{n}\delta\left(x_{i,j}=v_j\right)}{n}$$

$$P\left(x_{u,t}=v_t \middle| x_{u,j}=v_j\right) \approx \frac{\sum_{i=1}^{n}\delta\left(x_{i,t}=v_t\right)\delta\left(x_{i,j}=v_j\right)}{\sum_{i=1}^{n}\delta\left(x_{i,j}=v_j\right)}, \text{ где } \delta\left(x_{i,j}=v_j\right)=\begin{cases}1, x_{i,j}=v_j\\ 0, x_{i,j}\neq v_j\end{cases} \quad \textbf{(43)}$$

Из (42) и (43) следует, что мы принимаем решение $x_{u,j}=v_j$ при условии $x_{u,j_1}=v_{j_1},\ldots,x_{u,j_s}=v_{j_s}$ тогда и только тогда, когда

$$v_j=\arg\max\left\{\frac{\sum_{i=1}^{n}\delta\left(x_{i,j}=v\right)}{n}\frac{\sum_{i=1}^{n}\delta\left(x_{i,j_1}=v_{j_1}\right)\delta\left(x_{i,j}=v\right)}{\sum_{i=1}^{n}\delta\left(x_{i,j}=v\right)}\cdot\ldots\cdot\frac{\sum_{i=1}^{n}\delta\left(x_{u,j_s}=v_{j_s}\right)\delta\left(x_{i,j}=v\right)}{\sum_{i=1}^{n}\delta\left(x_{i,j}=v\right)}\middle| v\in V\right\}=$$

$$=\arg\max\left\{\frac{\sum_{i=1}^{n}\delta\left(x_{i,j_1}=v_{j_1}\right)\delta\left(x_{i,j}=v\right)\cdot\ldots\cdot\sum_{i=1}^{n}\delta\left(x_{u,j_s}=v_{j_s}\right)\delta\left(x_{i,j}=v\right)}{\left(\sum_{i=1}^{n}\delta\left(x_{i,j}=v\right)\right)^{s-1}}\middle| v\in V\right\} \quad \textbf{(44)}$$

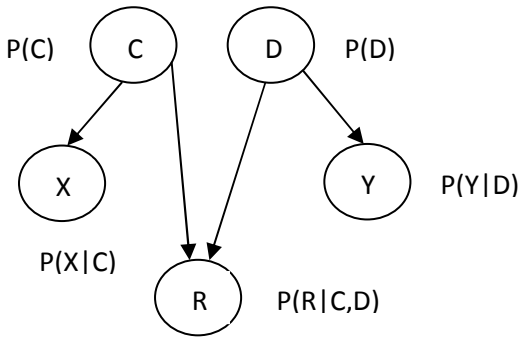
Если оценки безусловных и условных вероятностей из (43) получать по расписанию (например, ночью) и сохранять в базе данных, то для генерации рейтингов для одной транзакции в online режиме по формуле (44) потребуется O(V * m) арифметических операций, что не зависит от числа накопленных транзакций. Таким образом, разбиение процесса рекомендаций на отложенное периодическое обучение (вычисление и сохранение значений параметров) и прогнозирование рейтингов в реальном времени позволяет оптимизировать время выполнения операций.

### *Гибкая смешанная модель (Flexible Mixture Model)*

Идея этого метода основывается на работе [8] и заключается в следующем:

- Существует отдельный набор групп пользователей и отдельный набор групп товаров, вхождение в которые определяют значение рейтинга. Т.е. пользователи из группы i оценивают товары из группы j примерно одинаково.
- Пользователь или товар могут входить в несколько групп. Т.е. фильм может быть историческим, военным и психологическим одновременно.

Пусть пространство событий представляет собой подмножества из множества кортежей $\left\{\left(x_i,y_j,r_l\right)\middle| i\in\{1,\ldots,n\}, j\in\{1,\ldots,m\}, r\in\{1,\ldots,l\}\right\}$, случайная величина X принимает одно из значений $\{x_1,\ldots,x_n\}$ множества пользователей, случайная величина Y принимает одно из значений $\{y_1,\ldots,y_m\}$ множества товаров, а случайная величина R принимает одно из значений $\{r_1,\ldots,r_l\}$ множества возможных рейтингов. Т.е. множество элементарных событий – тройки пользователь, товар, рейтинг. Пусть также имеются ненаблюдаемые случайные величины: C принимает одно из значений $\{c_1,\ldots,c_I\}$ множества классов пользователей и D принимает одно из значений $\{d_1,\ldots,d_J\}$ множества классов товаров. Условная независимость случайных величин X, Y, R, C, D пусть описывается следующей Байесовской сетью:




Т.е. рейтинг непосредственно зависит только от групп, в которые входят пользователь и товар. Таким образом, совместное распределение вероятностей имеет вид:

$$P(X,Y,R,C,D) = P(C)P(D)P(X|C)P(Y|D)P(R|C,D) \quad \textbf{(45)}$$

$$P(X,Y,R) = \sum_C \sum_D P(C)P(D)P(X|C)P(Y|D)P(R|C,D) \quad \textbf{(46)}$$

Задача оценки неизвестных параметров $P(C),P(D),P(X|C),P(Y|D),P(R|C,D)$ определяется известным методом кластеризации – Expectation Maximization (EM). Суть метода заключается в итеративном выполнении двух шагов:

1. Шаг E: вычисление новой апостериорной вероятности вхождения в кластеры при помощи (45) и (44), где $P(C),P(D),P(X|C),P(Y|D),P(R|C,D)$ определены на прошлом шаге M или заданы произвольно на первом шаге:

$$P'(C,D|X,Y,R) = \frac{P(C)P(D)P(X|C)P(Y|D)P(R|C,D)}{\sum_C \sum_D P(C)P(D)P(X|C)P(Y|D)P(R|C,D)} \quad \textbf{(47)}$$

2. Шаг M: новые параметры $P'(C),P'(D),P'(X|C),P'(Y|D),P'(R|C,D)$ вычисляются по формулам, полученным при решении задачи максимизации логарифма функции правдоподобия, при фиксированном значении $P(C,D|X,Y,R)$, полученном на прошлом шаге E.

$$P'(C) = \frac{\sum_{(X,Y,R)} \sum_D P(C,D|X,Y,R)}{nm} \quad \textbf{(48)}$$

$$P'(D) = \frac{\sum_{(X,Y,R)} \sum_C P(C,D|X,Y,R)}{nm} \quad \textbf{(49)}$$

$$P'(X_i|C) = \frac{\sum_{(X,Y,R):X=X_i} \sum_D P(C,D|X,Y,R)}{nm \times P'(C)} \quad \textbf{(50)}$$

$$P'(Y_j|D) = \frac{\sum_{(X,Y,R):Y=Y_j} \sum_C P(C,D|X,Y,R)}{nm \times P'(D)} \quad \textbf{(51)}$$

$$P'(R_k|C,D)=\frac{\sum_{(X,Y,R):R=R_k} P(C,D|X,Y,R)}{\sum_{(X,Y,R)} P(C,D|X,Y,R)} \quad \textbf{(52)}$$

Алгоритм EM гарантирует монотонный рост функции правдоподобия, но не исключает возможности схождения в точке локального, а не глобального максимума, поэтому он выполняется несколько раз при различных начальных значениях параметров.

После выполнения алгоритма EM мы получаем оценку параметров модели $P(C),P(D),P(X|C),P(Y|D),P(R|C,D)$, что позволяет нам рассчитать прогноз неизвестного рейтинга товара Y нового пользователя $\bar{X}$ как условное математическое ожидание

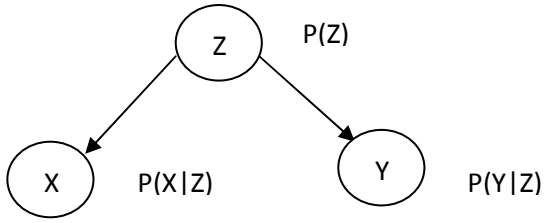
$$E(R|\bar{X},Y)=\sum_R R\cdot P(R|\bar{X},Y)=\sum_R R\cdot\frac{P(\bar{X},Y,R)}{\sum_{R'}P(\bar{X},Y,R')} \quad \textbf{(53)}$$

В формуле (53) мы используем выражение (46)

$P(\bar{X},Y,R)=\sum_C\sum_D P(C)P(D)P(\bar{X}|C)P(Y|D)P(R|C,D)$, в котором в правой части все параметры нам известны кроме $P(\bar{X}|C)$, т.к. пользователь $\bar{X}$ - новый и этот параметр для него не был рассчитан. Для его вычисления еще раз выполняется алгоритм кластеризации EM со всеми фиксированными значениями параметров кроме $P(\bar{X}|C)$.

### *Аспектная модель (Aspect Model)*

В аспектной модели ([11]) множество элементарных событий представляет собой пары $\{(x_i,y_j)|i\in\{1,...,n\},j\in\{1,...,m\}\}$. Генеральная совокупность состоит из наблюдений, представляющих собой пары совместно встречающихся транзакций и товаров $(x_i,y_j)$, интерпретируемые как «в i-ую транзакцию входит j-ый товар». С каждым наблюдением также связана ненаблюдаемая переменная класса $Z:\Omega\to\{z_1,...,z_k\}$, описывающая группы транзакций и товаров, «встречающихся» совместно. Ключевым допущением является условная независимость X и Y при известном Z, т.е. соответствующая байесовская сеть выглядит как:




$$P(X,Y,Z)=P(Z)P(X|Z)P(Y|Z) \quad \textbf{(54)}$$

Неизвестные параметры $P(Z),P(X|Z),P(Y|Z)$ определяются как обычно при помощи стандартной процедуры EM итеративной максимизации логарифма функции правдоподобия:

1. Шаг E: получение нового значения оценки $P'(Z|X,Y)$ при известных оценках $P(Z), P(X|Z), P(Y|Z)$, определенных на прошлом шаге M.

$$P'(Z|X,Y) = \frac{P(Z)P(X|Z)P(Y|Z)}{\sum_{i=1}^{k} P(Z_i)P(X|Z_i)P(Y|Z_i)} \quad \textbf{(55)}$$

2. Шаг M: определение новых оценок $P'(Z), P'(X|Z), P'(Y|Z)$ как решение задачи максимизации функции правдоподобия при фиксированном значении $P(Z|X,Y)$, определенном на прошлом шаге E.

$$P'(Z) = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} n(X_i,Y_j)P(Z|X_i,Y_j)}{\sum_{i=1}^{n}\sum_{j=1}^{m} n(X_i,Y_j)} \quad \textbf{(56)}$$

$$P'(X|Z) = \frac{\sum_{j=1}^{m} n(X,Y_j)P(Z|X,Y_j)}{\sum_{i=1}^{n}\sum_{j=1}^{m} n(X_i,Y_j)} \quad \textbf{(57)}$$

$$P'(Y|Z) = \frac{\sum_{i=1}^{n} n(X_i,Y)P(Z|X_i,Y)}{\sum_{i=1}^{n}\sum_{j=1}^{m} n(X_i,Y_j)} \quad \textbf{(58)}$$

В выражениях (56)-(58) n(X,Y) обозначает число раз, когда пара (X,Y) встречалась в данных совместно, принимает значение 0 или 1.

После получения оценки $P(Z), P(X|Z), P(Y|Z)$ мы можем оценить вероятность того, что (новый) пользователь $\bar{X}$ приобретет товар Y:

$$P(Y|\bar{X}) = \frac{P(\bar{X},Y)}{P(\bar{X})} = \frac{\sum_{l=1}^{k} P(\bar{X},Y,Z_l)}{\sum_{j=1}^{m}\sum_{l=1}^{k} P(\bar{X},Y_j,Z_l)} = \frac{\sum_{l=1}^{k} P(Z_l)P(\bar{X}|Z_l)P(Y|Z_l)}{\sum_{j=1}^{m}\sum_{l=1}^{k} P(Z_l)P(\bar{X}|Z_l)P(Y_j|Z_l)} \quad \textbf{(59)}$$

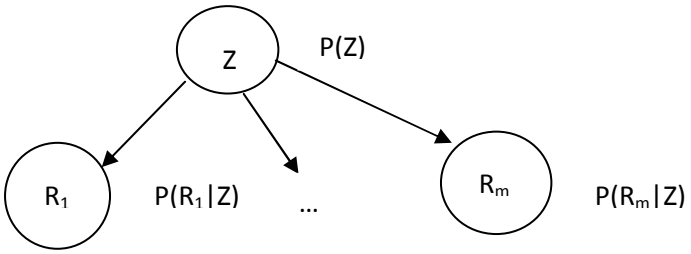
Значения $P(\bar{X}|Z_l)$ нам неизвестны, т.к. пользователь $\bar{X}$ новый. Для их вычисления еще раз выполняется алгоритм EM со всеми фиксированными значениями параметров кроме $P(\bar{X}|Z_l)$. В результате пользователю $\bar{X}$ рекомендуются товары Y с максимальной условной вероятностью $P(Y|\bar{X})$.

### *Смешанная мультиномиальная модель (Mixture of Multinomials Model)*

В смешанной мультиномиальной модели ([19]) мы предполагаем, что существуют K типов пользователей, причем пользователи одного типа присваивают схожие рейтинги одинаковым товарам. Таким образом, пользователей можно сгруппировать в кластеры исходя из выставленных ими значений рейтингов товарам. Мы предполагаем, что генеральная совокупность

представляет собой множество транзакций, каждая из которых представлена набором случайных переменных $R_{j_1^i},...,R_{j_{s_i}^i}$, описывающих рейтинги, выставленные товарам $j_1^i,...,j_{s_i}^i \in \{1,...,m\}$ в этой i-ой транзакции, и ненаблюдаемой переменной класса Z. Рейтинг каждого товара может принимать одно из значений от 1 до V, а переменная класса – от 1 до K. Далее мы постулируем, что при известном значении класса, рейтинги различных товаров независимы, т.е.

$$P\left(R_{j_1},...,R_{j_s},Z\right)=P(Z)\prod_{p=1}^{s}P\left(R_{j_p}\middle|Z\right) \quad \textbf{(60)}$$




Таким образом, совместная вероятность подмножества рейтингов в i-ой транзакции равна

$$P\left(R_{j_1},...,R_{j_s}\right)=\sum_{c=1}^{K}P(Z_c)\prod_{p=1}^{s}P\left(R_{j_p}\middle|Z_c\right) \quad \textbf{(61)}$$

Параметрами модели являются:

$\theta_z := P(Z=z), z\in\{1,...,K\}$ - априорная вероятность принадлежности транзакции к классу z,

$\beta_{v,y,z} := P\left(R_y=v\middle|Z=z\right), v\in\{1,...,V\}, y\in\{1,...,M\}, z\in\{1,...,K\}$ - условная вероятность того, что y-ому товару был присвоен рейтинг v при условии, что транзакция принадлежит классу z,

$\varphi_{i,z} := P\left(Z=z\middle|X_i\right)=P\left(Z=z\middle|R_{j_1^i}=x_{i,j_1^i},...,R_{j_{s_i}^i}=x_{i,j_{s_i}^i}\right), i\in\{1,...,N\}, z\in\{1,...,K\}$ - апостериорная вероятность принадлежности транзакции к классу z при известном профиле транзакции, описываемом выставленными в этой транзакции рейтингами товарам $j_1^i,...,j_{s_i}^i \in \{1,...,m\}$.

Логарифм совместной вероятности всех транзакций $X_1,...,X_N$ равен:

$$\begin{aligned}
&\ln P\left(X_1,...,X_N\right)=\sum_{i=1}^{N}\ln P\left(X_i\right)\underbrace{=}_{\forall z\in\{1,...,K\}}\sum_{i=1}^{N}\ln\frac{P\left(X_i,Z=z\right)}{P\left(Z=z\middle|X_i\right)}=\\
&=\sum_{i=1}^{N}\underbrace{\sum_{z=1}^{K}\underbrace{P\left(Z=z\middle|X_i\right)}_{=\varphi_{i,z}}}_{=1\forall i}\ln\frac{P\left(X_i,Z=z\right)}{P\left(Z=z\middle|X_i\right)}=\\
&=\sum_{i=1}^{N}\sum_{z=1}^{K}\varphi_{i,z}\ln\frac{P\left(X_i\middle|Z=z\right)\underbrace{P\left(Z=z\right)}_{=\theta_z}}{\varphi_{i,z}}=\\
&=\sum_{i=1}^{N}\sum_{z=1}^{K}\varphi_{i,z}\ln\left(\theta_z P\left(X_i\middle|Z=z\right)\right)-\sum_{i=1}^{N}\sum_{z=1}^{K}\varphi_{i,z}\ln\varphi_{i,z}
\end{aligned} \quad \textbf{(62)}$$

Далее, из (60) следует:

$$P\left(X_i\middle|Z=z\right)=P\left(R_{j_1^i}=x_{i,j_1^i},...,R_{j_{s_i}^i}=x_{i,j_{s_i}^i}\middle|Z=z\right)=\prod_{j\in\left\{j_1^i,...,j_{s_i}^i\right\}}P\left(R_j=x_{i,j}\middle|Z=z\right)=$$

$$=\prod_{y=1}^{M}\prod_{v=1}^{V}\left[P\left(R_j=x_{i,j}\middle|Z=z\right)\right]^{\delta\left(x_{i,y},v\right)}=\prod_{y=1}^{M}\prod_{v=1}^{V}\beta_{v,y,z}^{\delta\left(x_{i,y},v\right)} \quad \textbf{(63)}$$

где $\delta\left(x_{i,y},v\right)$ равно 1, если в i-ой транзакции y-ому товару был выставлен рейтинг v. Это значение равно 0 в противном случае. Подставляя (63) в (62) получаем:

$$\ln P\left(X_1,...,X_N\right)=\sum_{i=1}^{N}\sum_{z=1}^{K}\phi_{i,z}\ln\left(\theta_z\prod_{y=1}^{M}\prod_{v=1}^{V}\beta_{v,y,z}^{\delta\left(x_{i,y},v\right)}\right)-\sum_{i=1}^{N}\sum_{z=1}^{K}\phi_{i,z}\ln\phi_{i,z} \quad \textbf{(64)}$$

Задача максимизации (64) относительно параметров модели решается методом Expectation Maximization:

**Шаг E**: определение новых значений параметра

$$\phi'_{i,z}:=P\left(Z=z\middle|X_i\right)=\frac{P\left(X_i\middle|Z=z\right)P\left(Z=z\right)}{\sum_{c=1}^{K}P\left(X_i\middle|Z=c\right)P\left(Z=c\right)}=\frac{\theta_z\prod_{y=1}^{M}\prod_{v=1}^{V}\beta_{v,y,z}^{\delta\left(x_{i,y},v\right)}}{\sum_{c=1}^{K}\theta_c\prod_{y=1}^{M}\prod_{v=1}^{V}\beta_{v,y,c}^{\delta\left(x_{i,y},v\right)}} \quad \textbf{(65)}$$

на основании значений параметров $\theta_z,\beta_{v,y,z}$ , определенных на предыдущем шаге M.

**Шаг M**: определение новых значений параметров $\theta_z^{'},\beta'_{v,y,z}$ на основании значений параметров $\phi_{i,z}$ , определенных на прошлом шаге E. Эти параметры определяются максимизацией функции правдоподобия (64) относительно $\theta_z,\beta_{v,y,z}$ при фиксированных значениях $\phi_{i,z}$ :

$$\theta'_z:=\frac{\sum_{i=1}^{N}\phi_{i,z}}{\sum_{c=1}^{K}\sum_{i=1}^{N}\phi_{i,c}} \quad \textbf{(66)}$$

$$\beta'_{v,y,z}:=\frac{\sum_{i=1}^{N}\phi_{i,z}\delta\left(x_{i,y},v\right)}{\sum_{c=1}^{V}\sum_{i=1}^{N}\phi_{i,z}\delta\left(x_{i,y},c\right)} \quad \textbf{(67)}$$

После определения параметров модели вычисляется условная вероятность присвоения новым пользователем $X_u$ рейтинга со значением v j-ому товару:

$$P\left(R_j=v\middle|X_u\right)=\sum_{z=1}^{K}P\left(R_j=v,Z=z\middle|X_u\right)=\sum_{z=1}^{K}P\left(Z=z\middle|X_u\right)P\left(R_j=v\middle|Z=z,X_u\right)=$$

$$=\sum_{z=1}^{K}P\left(Z=z\middle|X_u\right)P\left(R_j=v\middle|Z=z\right)=\sum_{z=1}^{K}\frac{P\left(X_u\middle|Z=z\right)P\left(Z=z\right)}{\sum_{c=1}^{K}P\left(X_u\middle|Z=c\right)P\left(Z=c\right)}\beta_{v,j,z}= \quad \textbf{(68)}$$

$$=\sum_{z=1}^{K}\frac{\theta_z\prod_{y=1}^{M}\prod_{p=1}^{V}\beta_{p,y,z}^{\delta\left(x_{u,y},p\right)}}{\sum_{c=1}^{K}\theta_c\prod_{y=1}^{M}\prod_{p=1}^{V}\beta_{p,y,z}^{\delta\left(x_{u,y},p\right)}}\beta_{v,j,z}$$

Прогноз значения рейтинга j-ого товара со стороны исследуемого u-ого пользователя может осуществляться одним из следующих способов:

$$\hat{x}_{u,j} := \sum_{v=1}^{K} vP\left(R_j = v \middle| X_u\right), \quad \textbf{(69)}$$

в этом случае мы минимизируем средний квадрат ошибки (Minimum Square Error Estimator)

$$\hat{x}_{u,j} := median\left\{P\left(R_j = v \middle| X_u\right) \middle| v \in \{1,...,K\}\right\}, \quad \textbf{(70)}$$

в этом случае минимизируется средняя абсолютная ошибка (Minimum Abslout Error Estimator)

$$\hat{x}_{u,j} := \arg\max\left\{P\left(R_j = v \middle| X_u\right) \middle| v \in \{1,...,K\}\right\}, \quad \textbf{(71)}$$

в этом случае минимизируется вероятность ошибки (Maximum Aposteriori Estimator).

### *Совместная кластеризация товаров и транзакций*

Идея совместной кластеризации по товарам и транзакциям основана на работе [12] и заключается в аппроксимации матрицы рейтингов с помощью матрицы, состоящей из средних значений рейтингов по кластерам товаров и транзакций. Рассмотрим отображения $\rho : \{1,...,n\} \to \{1,...,k\}$ и $\gamma : \{1,...,m\} \to \{1,...,l\}$, ставящие в соответствие каждой транзакции и каждому товару соответствующий кластер транзакций или товаров, где k-число кластеров транзакций, а l-число кластеров товаров. В качестве прогноза неизвестного рейтинга j-ого товара в i-ой транзакции используется среднее значение известных рейтингов из соответствующих кластеров $\rho(i)$ и $\gamma(j)$. Чтобы учесть смещение в рейтингах для j-ого товара в i-ой транзакции относительно средних значений в соответствующих кластерах аппроксимирующая матрицы имеет вид:

$$\hat{x}_{i,j} = x^{COC}_{\rho(i),\lambda(j)} + \left(x^{R}_{i} - x^{RC}_{\rho(i)}\right) + \left(x^{C}_{j} - x^{CC}_{\gamma(j)}\right), \text{ где} \quad \textbf{(72)}$$

$$x^{COC}_{\rho(i),\gamma(j)} := \frac{\sum_{p=1,\rho(p)=\rho(i)}^{n} \sum_{t=1,\gamma(t)=\gamma(j)}^{m} x_{p,t} \cdot n(p,t)}{\sum_{p=1,\rho(p)=\rho(i)}^{n} \sum_{t=1,\gamma(t)=\gamma(j)}^{m} n(p,t)} \quad \textbf{(73)}$$

- средний рейтинг товаров из $\gamma(j)$ кластера, входящих в транзакции из $\rho(i)$ кластера.

$$x^{R}_{i} := \frac{\sum_{t=1}^{m} x_{i,t} \cdot n(i,t)}{\sum_{t=1}^{m} n(i,t)} \quad \textbf{(74)}$$

- средней рейтинг по всем товарам i-ой транзакции

$$x^{C}_{j} := \frac{\sum_{p=1}^{n} x_{p,j} \cdot n(p,j)}{\sum_{p=1}^{n} n(p,j)} \quad \textbf{(75)}$$

- средний рейтинг j-ого товара по всем транзакциям

$$x^{RC}_{\rho(i)} := \frac{\sum_{p=1,\rho(p)=\rho(i)}^{n} \sum_{t=1}^{m} x_{p,t} \cdot n(p,t)}{\sum_{p=1,\rho(p)=\rho(i)}^{n} \sum_{t=1}^{m} n(p,t)} \quad \textbf{(76)}$$

- средний рейтинг по всем товарам из транзакций, входящих в $\rho(i)$ -ый кластер

$$x^{CC}_{\gamma(j)} := \frac{\sum_{p=1}^{n}\sum_{t=1,\gamma(t)=\gamma(j)}^{m} x_{p,t}\cdot n(p,t)}{\sum_{p=1}^{n}\sum_{t=1,\gamma(t)=\gamma(j)}^{m} n(p,t)} \quad \textbf{(77)}$$

средний рейтинг товаров из $\gamma(j)$ -ого кластера по всем транзакциям.

В выражениях (73)-(77) $n(p,t)$ равно 1 если рейтинг p-ого товара в t-ой транзакции известен и 0 в противном случае. Этот множитель служит для усреднения только по известным рейтингам. Выражение (72), таким образом, формирует прогноз неизвестного рейтинга j-ого товара в i-ой транзакции как среднее по рейтингам товаров и транзакций, входящих в те же кластера $\rho(i)$ и $\gamma(j)$ с коррекцией на отклонение среднего рейтинга j-ого товара от среднего рейтинга кластера $\gamma(j)$ товаров и отклонение среднего рейтинга i-ой транзакции от среднего рейтинга кластера $\rho(i)$ транзакций.

Процесс кластеризации заключается в нахождении функций $\rho(i)$ и $\gamma(j)$, минимизирующих квадратичную форму $\sum_{i=1}^{n}\sum_{j=1}^{m}\left(x_{i,j}-\hat{x}_{i,j}\right)^2$, **(78)**

где $\hat{x}_{i,j}$ согласно (72) зависит от $\rho(i)$ и $\gamma(j)$. Ниже приведен итеративный алгоритм, сходящийся согласно [13] к локальному минимуму (78).

***Алгоритм определения параметров модели (обучение)***

**Входные данные**: матрица рейтингов X, матрица N (состоящая из 1 или 0 в зависимости от того известен ли соответствующий рейтинг или нет), число кластеров k и l.

**Выходные данные**: функции $\rho(i)$ и $\gamma(j)$ и средние $x^{COC}_{\rho(i),\lambda(j)}$, $x^{R}_{i}$, $x^{RC}_{\rho(i)}$, $x^{C}_{j}$, $x^{CC}_{\gamma(j)}$.

**Метод**:

1. Задать $\rho(i)$ и $\gamma(j)$ случайным образом.
2. Вычислить $x^{COC}_{\rho(i),\lambda(j)}$, $x^{R}_{i}$, $x^{RC}_{\rho(i)}$, $x^{C}_{j}$, $x^{CC}_{\gamma(j)}$ по формулам (61)-(65).
3. Решить задачу оптимизации (минимум по строке):

$$\rho(i) := \arg\min\left\{\sum_{j=1}^{m} n(i,j)\cdot\left(x_{i,j}-\left[x^{COC}_{g,\gamma(j)}+\left(x^{R}_{i}-x^{RC}_{g}\right)+\left(x^{C}_{j}-x^{CC}_{\gamma(j)}\right)\right]\right)^2 \middle| g\in\{1,\ldots k\}\right\}$$

4. Решить задачу оптимизации (минимум по столбцу):

$$\gamma(j) := \arg\min\left\{\sum_{i=1}^{n} n(i,j)\cdot\left(x_{i,j}-\left[x^{COC}_{\rho(i),h}+\left(x^{R}_{i}-x^{RC}_{\rho(i)}\right)+\left(x^{C}_{j}-x^{CC}_{h}\right)\right]\right)^2 \middle| h\in\{1,\ldots l\}\right\}$$

5. Вычислить $\hat{x}_{i,j}$ по формуле (72) и $\sum_{i=1}^{n}\sum_{j=1}^{m}\left(x_{i,j}-\hat{x}_{i,j}\right)^2$. Если уменьшение $\sum_{i=1}^{n}\sum_{j=1}^{m}\left(x_{i,j}-\hat{x}_{i,j}\right)^2$ меньше заранее заданного числа, прекратить, в противном случае – перейти к шагу 2.

Алгоритм определения параметров модели выполняется в отложенном режиме.

***Прогнозирование***

Прогнозирование неизвестного рейтинга j-ого товара в новой (не участвующей в обучении) u-ой транзакции производится по формуле (72) $\hat{x}_{u,j} = x^{COC}_{\rho(u),\lambda(j)} + \left(x^R_u - x^{RC}_{\rho(u)}\right) + \left(x^C_j - x^{CC}_{\gamma(j)}\right)$, где привязка u-ой транзакции к кластеру $\rho(u)$ осуществляется в форме решения задачи оптимизации

$$\rho(u) := \arg\min\left\{\sum_{j=1}^{m} n(u,j)\cdot\left(x_{u,j} - \left[x^{COC}_{g,\gamma(j)} + \left(x^R_u - x^{RC}_g\right) + \left(x^C_j - x^{CC}_{\gamma(j)}\right)\right]\right)^2 \middle| g \in \{1,...k\}\right\}.$$

## *Усреднение по кластерам транзакций*

Идея этого подхода заключается в группировке транзакций, а затем вычисления неизвестного рейтинга в исследуемой транзакции как средневзвешенное из рейтингов этого товара в K наиболее близких транзакциях их самого близкого кластера. Сама идея аналогична методу K ближайших соседей, в котором для исследуемой транзакции мы определяли K наиболее похожих на нее транзакций, а рейтинги вычислялись как наиболее вероятные среди этих K элементов.

В статье [14] предложен следующий подход:

1. Методом K-Means исходное множество транзакций разбивается на N кластеров. Т.е. на первом шаге случайным образом выбираются N транзакций, которые служат центрами соответствующих кластеров. Затем каждая транзакция приписывается к кластеру, расстояние до центра которого минимально. Затем обновляются значения центров кластеров: ими становятся средние значения входящих в них транзакций. Процедура повторяется до тех пор пока уменьшение расстояния между центрами кластеров и входящих в эти кластера транзакций не станет достаточно маленьким. В качестве меры близости между транзакциями принимается корреляция Пирсона:

$$s_{trans}\left(X_u, X_i\right) := \frac{\sum^{m}_{k=1;x_{u,k},x_{i,k}\ \text{определены}}\left(x_{u,k} - \overline{x}_u\right)\left(x_{i,k} - \overline{x}_i\right)}{\sqrt{\sum^{m}_{k=1;x_{u,k},x_{i,k}\ \text{определены}}\left(x_{u,k} - \overline{x}_u\right)^2 \sum^{m}_{k=1;x_{u,k},x_{i,k}\ \text{определены}}\left(x_{i,k} - \overline{x}_i\right)^2}} \quad \textbf{(79)}$$

   В итоге мы получаем соответствие $T : \{1,...,n\} \to \{1,...,N\}$ между множеством транзакций и множеством кластеров $C_1,...,C_N$.

2. В исходных данных проставляются пропущенные значения рейтингов как среднее в кластере значение отклонения рейтинга товара от среднего рейтинга в транзакции. Т.е.

$$\tilde{x}_{k,l} := \overline{x}_k + \frac{\sum_{i \in T(k)}\left(x_{i,l} - \overline{x}_i\right)}{\left|C_{T(k)}\right|} \quad \textbf{(80)}$$

   Далее, в качестве рейтинга j-ого товара в i-ой транзакции мы будем рассматривать

$$x_{i,j} := \begin{cases} x_{i,j}, \textit{рейтинг есть} \\ \tilde{x}_{i,j}, \textit{рейтинга нет} \end{cases} \quad \textbf{(81)}$$

3. Для исследуемой транзакции $X_u$ определяется кластер к которой она принадлежит как кластер, расстояние до центра которого минимально. Т.е. сначала определяется расстояние между транзакцией $X_u$ и кластерами $C_1$,…, $C_N$:

$$s_{cluster}(X_u, C_i) := \frac{\sum_{k=1}^{m}\left(x_{u,k} - \overline{x}_u\right)\frac{\sum_{j\in C_i}\left(x_{j,k} - \overline{x}_j\right)}{|C_i|}}{\sqrt{\sum_{k=1}^{m}\left(x_{u,k} - \overline{x}_u\right)^2 \sum_{i=1}^{m}\left(\frac{\sum_{j\in C_i}\left(x_{j,k} - \overline{x}_j\right)}{|C_i|}\right)^2}} \quad \textbf{(82)}$$

А затем определяется $T(u) := \arg\min\left\{s_{cluster}(X_u, C_i) \middle| i \in \{1,...,N\}\right\}$ **(83)**

4. Для $\lambda \in [0,1]$ определяем меру доверия к рейтингу j-ого товара в i-ой транзакции как

$$w_{i,j} := \begin{cases} 1-\lambda, \text{рейтинг есть} \\ \lambda, \text{рейтинга нет} \end{cases}, \quad \textbf{(84)}$$

в зависимости от того был ли рейтинг проставлен пользователем или вычислен усреднением по кластеру при помощи формулы (80).

5. После того, как мы с помощью (82), (83) определили кластер T(u) для исследуемой транзакции, проставляем для нее неизвестные рейтинги по формулам (80), (81).
6. Далее определяем меру близости между $X_u$ и транзакциями из ее кластера T(u):

$$s_{trans-cluster}(X_u, X_i) := \frac{\sum_{k=1}^{m}\left(x_{u,k} - \overline{x}_u\right)\left(x_{i,k} - \overline{x}_i\right)}{\sqrt{\sum_{k=1}^{m}\left(x_{u,k} - \overline{x}_u\right)^2 \sum_{i=1}^{m}\left(x_{i,k} - \overline{x}_i\right)^2}}, i \in T(u) \quad \textbf{(85)}$$

Выбираем K транзакций из кластера T(u) с наименьшими расстояниями до $X_u$. Обозначим это множество $\{u_1,...,u_K\}$.

7. Определяем неизвестный рейтинг p-ого товара как

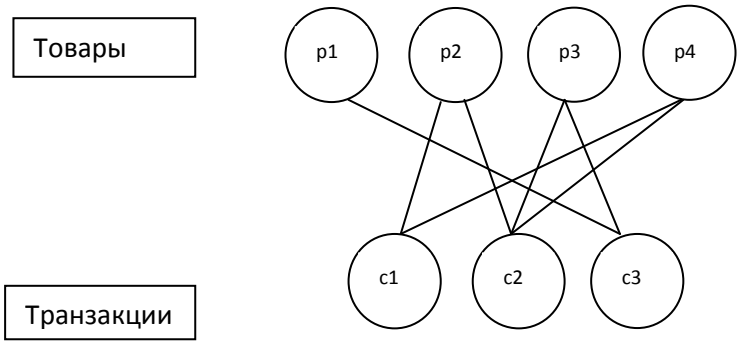
$$\hat{x}_{u,j} := \overline{x}_u + \frac{\sum_{i=1}^{K} w_{u_i,j}\, s_{trans-cluster}(X_u, X_{u_i})\left(x_{u_i,j} - \overline{x}_{u_i}\right)}{\sum_{i=1}^{K} w_{u_i,j}\left|s_{trans-cluster}(X_u, X_{u_i})\right|} \quad \textbf{(86)}$$

К преимуществам описанного алгоритма по сравнению с фильтрацией по транзакциям относятся следующие факты:

- Меру близости к анализируемой транзакции надо вычислять в реальном времени только для транзакций из одного кластера, а не для всех имеющихся транзакций.
- Пропущенные рейтинги в имеющихся транзакциях заменяются на средние по кластеру, что важно при анализе разреженных данных (например, рейтинги кинофильмов).

### *Транзитивные ассоциативные сети*

В статье [15] описан подход, основанный на построении транзитивных связей между транзакциями и товарами. Идея этого подхода может быть проиллюстрирована на следующем примере. Допустим, что пользователи c1 и c2 приобрели товар p1, а пользователи c2 и c3 приобрели товар p2. Стандартные алгоритмы совместной фильтрации свяжут пользователей c1 и c2, а также пользователей c2 и c3, но не свяжут c1 и c3. Для получения транзитивных связей между транзакциями используется граф, узлы которого состоят из двух частей – транзакции и товары, а дуги связывают транзакции с входящими в них товарами.




Соответствующая матрица связности имеет вид:

$$\begin{array}{c} \\ c1 \\ c2 \\ c3 \end{array}\begin{array}{c} \begin{array}{cccc} p1 & p2 & p3 & p4 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

Предположим, что наша задача заключается в формировании рекомендации товаров для пользователя в транзакции c1. Совместная фильтрация построит рекомендации на основании сходства между транзакцией c1 с c2 и c3. Сходство c1 с c2 очевидно, так как в обе транзакции входят товары p2 и p4. В результате для транзакции c1 будет рекомендован товар p3 потому что он был приобретен в транзакции c2. Сходство между транзакциями c1 и c3 стандартным алгоритмом совместной фильтрации найдено не будет, и поэтому товар p1, входящий в c3, не будет рекомендован для c1.

Для учета транзитивных связей между транзакциями мы рассматриваем все пути между ними с длиной не больше заданного числа M. Стандартные алгоритмы совместной фильтрации учитывают пути длиной 3, например c1-p2-c2 или c1-p4-c2. Интуитивно понятно, что чем больше число различных путей, соединяющих два узла, тем сильнее связь между ними. Также интуитивно понятно, что чем длинней путь, связывающий два узла, тем слабее связь между ними. В [15] мера близости между транзакцией c и товаром p определяется как сумма весов $\alpha(c,p)$ всех различных путей, соединяющих c и p. Вес каждого пути определяется как $\rho^x$, где $\rho \in (0,1)$, а $x \le M$ - длина пути. Таким образом, для нашего примера при $\rho = \frac{1}{2}$ мера близости между c1 и p3 будет равна $\alpha(c_1, p_3) = 2\left(\frac{1}{2}\right)^3 = 0,25$ с учетом двух различных путей длины 3: c1-p2-c2-p3 и c1-p4-c2-p3. А мера близости между c1 и p1 будет равна $\alpha(c_1, p_1) = \left(\frac{1}{2}\right)^5 = 0,03125$ для единственного пути длиной 5: c1-p2-c2-p3-c3-p1. Таким образом, рекомендательная сила товара p1 будет значительно меньше, чем p3, но в отличие от стандартных алгоритмов совместной фильтрации, она будет все же отлична от нуля.

Пусть дана матрица связности графа A, т.е. матрица описывающая связи между продуктами и транзакциями. Тогда по индукции можно показать, что матрица достижимости за длину пути x (т.е. матрица A(x), элементами которой являются число различных путей длины x между продуктами и транзакциями) равна:

$$A(x)=\begin{cases} A, x=1 \\ A\cdot A^t\cdot A(x-2), x=3,5,7,... \end{cases}=\left(A\cdot A^t\right)^{\frac{x-1}{2}}\cdot A \quad \textbf{(87)}$$

Соответственно, матрица $W(X)$ весов всех путей длины x, связывающих товары и транзакции будет равна

$$W(x)=\rho^x A(x)=\rho^x\left(A\cdot A^t\right)^{\frac{x-1}{2}}\cdot A \quad \textbf{(88)}$$

Так как матрица $\alpha(M)$ степеней сходства между товарами и транзакциями определена как сумма весов всех связывающих их путей с длиной, не превышающей M, то из (76) следует:

$$\alpha(M)=\sum_{x=1}^{\frac{M+1}{2}}W(2x-1)=A\sum_{x=1}^{\frac{M+1}{2}}\rho^{2x-1}\left(A\cdot A^t\right)^{x-1} \quad \textbf{(89)}$$

Процесс выработки рекомендаций можно разделить на две фазы:

1. В отложенном режиме считаются матрицы, элементами которых являются количество различных путей длины x, связывающие товары с товарами: $A(x)=\left(A^tA\right)^{\frac{x}{2}}$, матрицы суммы весов путей длины x: $W(x)=\rho^x\left(A^tA\right)^{\frac{x}{2}}$, матрицы степеней сходства между товарами $\alpha(M-1)=\sum_{x=1}^{\frac{M-1}{2}}\rho^{2x}\left(A^tA\right)^x$ как сумма весов всех путей между ними длины меньше M.
2. В реальном времени степень сходства между новой исследуемой транзакцией и исследуемым товаром равна сумме весов всех путей от всех товаров, входящих в исследуемую транзакцию, к исследуемому товару с поправкой на коэффициент $\rho$ (длина пути увеличиалсь на 1), т.е. сумме степеней сходства между исследуемым товаром и товарами, входящими в исследуемую транзакцию, умноженную на $\rho$. Эта величина равна суммам элементов матрицы $\rho\alpha(M-1)$, соответствующих исследуемому товару и товарам, входящим в исследуемую транзакцию. Рекомендуются товары, степень сходства которых с исследуемой транзакцией максимальна.

### *Ассоциативные правила*

Выявление ассоциативных правил – это процесс определения частых наборов товаров, приобретаемых совместно в транзакциях. На основании этих частых наборов формируются правила вида «если А и Б, то В с вероятностью х». На основании сформированных правил рекомендуются (проставляются рейтинги) товарам, которые встречаются в правой части правил, если товары из левой части уже есть в транзакции. Таким образом, подход, основанный на ассоциативных правилах аналогичен фильтрации по товарам (рекомендуются товары, которые часто приобретаются совместно с уже заказанными товарами) особенно в варианте «частоты попарного вхождения». Разница заключаются в том, что в алгоритме ассоциативных правил правила формируются в отложенном режиме по расписанию, а поиск рекомендаций осуществляется в реальном времени на основании уже полученных правил. В фильтрации по товарам мы вычисляем веса, описывающие частоту попарного вхождения для каждого вектора товара в реальном времени в процессе формирования рекомендаций, что значительно более трудоемко. С другой стороны, недостатком алгоритма ассоциативных правил является то, что не для каждого набора заказанных товаров в анализируемой транзакции существует правило с

достаточной поддержкой и достоверностью. Так, например, для существования правил вида «А, Б -> В» для каждой пары товаров А и Б необходимо существование $3C_n^3$ частых набора (существует $C_n^3$ подмножеств из 3 элементов в множестве из n элементов, для каждого подмножества из 3 элементов можно сформировать 3 правила в зависимости от того, какой элемент будет в правой части этого правила). Для n=1000, число частых наборов в этом случае должно быть 498 501 000, что при ограничении на частоту набора в 10 транзакциях приводит к необходимости хранения не меньше 5 млрд. транзакций, что является очень серьезным требованием. Эта проблема решается поиском правил, содержащих только один элемент в левой части, т.е. типа «если А, то Б», для каждого товара А, уже имеющегося в транзакции. Полученным правым частям проставляется рейтинг в зависимости от вероятности правила. Этот подход еще в большей степени соответствует фильтрации по товарам с мерой близости векторов товаров в виде частоты попарного вхождения.

## Выбор критерия сравнения алгоритмов

Проблема выбора подходящего параметра измерения точности осложняется огромным разнообразием параметров, которые использовались для количественной оценки точности работы РС-систем в опубликованных исследованиях. Отсутствие стандартизации в этом вопросе наносит вред прогрессу в этой области знаний, относящейся к развитию рекомендационных систем на базе совместной фильтрации. Не имея стандартного параметра в качестве меры точности в этой сфере, исследователи продолжают вводить новые единицы измерения для оценки своих систем. При таком разнообразии используемых оценочных параметров становится сложно сравнивать результаты одного опубликованного исследования с результатами другого. В результате, становится тяжело интегрировать эти разные публикации в единое целое, чтобы выработать какие-либо общие знания и понятия относительно качества работы алгоритмов РС.

Растёт понимание того, что хорошая точность рекомендаций сама по себе не удовлетворяет потребностям пользователей РС-системы и не характеризует эффективность её работы. РС-системы должны предоставлять не только точные, но и полезные рекомендации. Например, РС-система могла бы достичь высокой точности исключительно за счёт формирования прогнозов для легко предсказуемых объектов, но это те объекты, относительно которых пользователи менее всего нуждаются в рекомендациях. Далее, система, которая всегда рекомендует очень популярные объекты, может гарантировать, что пользователям понравится б**о**льшая часть рекомендуемых объектов, но простой показатель популярности мог бы делать то же самое.

### Точность

Тестирование на точность заключается в случайном разделении транзакций на обучающее и тестовое множество (в пропорции 90%-10%, 80%-20%). Транзакции в обучающем множестве служат для оценки рейтингов товаров из транзакций из тестового множества. Товары из каждой тестовой транзакции случайно разделяются на две группы: «известные» и «неизвестные». На основании рейтингов группы «известных» товаров строятся рейтинги для группы «неизвестных» товаров - на основании данных из множества обучающих транзакций. В качестве меры точности прогноза служит средняя абсолютная ошибка прогноза рейтингов MAE (Mean Absolute Error):

$$MAE = \frac{\sum_{i \in I} \sum_{k \in K_i} \left| \overline{x}_{u,k} - x_{u,k} \right|}{\sum_{i \in I} \# K_i}, \quad \textbf{(90)}$$

где I – индексы тестового множества транзакций, а $K_i$ - множество «неизвестных» товаров в каждой тестовой транзакции.

### Покрытие

Покрытие (зона действия, охват) рекомендательной системы – это измерение области объектов в системе, по которым РС может формировать прогнозы или выдавать рекомендации. Системы с низким покрытием могут быть менее значимы для пользователей, так как они будут ограничены в принятии решений только теми из них, в которых РС будут способны им помочь. Общепринятой мерой покрытия является доля от общего числа объектов, для которых могут быть выработаны прогнозы. Самый простой способ измерить покрытие такого рода – это выбрать произвольную выборку пар пользователь/объект, запросить прогноз для каждой пары и измерить процент тех, в отношении которых был сделан прогноз. Покрытие должно замеряться в комбинации с точностью, так чтобы РС-системы не склонялись к увеличению покрытия за счёт выработки фиктивных прогнозов для каждого объекта.

### Скорость обучаемости системы

Рекомендательные системы на основе совместной фильтрации имеют в своём составе алгоритмы по самообучению, которые функционируют на статистических моделях. В итоге результаты их работы варьируются в зависимости от объёма доступной для обучения информации. По мере увеличения количества обучающей информации качество прогнозов или рекомендаций должно расти. Различные алгоритмы выработки рекомендаций могут достичь приемлемого качества рекомендаций с разной скоростью. Некоторым алгоритмам может понадобиться только небольшой объём информации, чтобы начать вырабатывать приемлемые рекомендации, в то время как другим может понадобиться достаточной большой объём. В РС-системах рассматривается 3 разных скорости накопления знаний: общая скорость обучаемости, скорость обучаемости по 1 объекту, и скорость обучаемости по 1 пользователю.

Общая скорость обучаемости РС-системы – это качество рекомендаций, выраженное как функция от общего числа рейтингов в системе (или общего числа пользователей системы). Скорость обучаемости по объекту – это качество рекомендаций относительно определённого объекта, выраженное как функция от числа рейтингов, имеющихся у определённого объекта. Также, скорость обучаемости по 1 пользователю – это качество рекомендаций для определённого пользователя, выраженное как функция от числа рейтингов, который конкретный пользователь ввёл в систему.

### Степень новизны

Можно представить себе РС-систему, выдающую очень точные рекомендации и имеющую достаточное покрытие, и тем не менее бесполезную для практических целей. Например, РС система овощного магазина может предлагать купить картошку любому покупателю, который ее ещё не выбрал. Статистически такая РС предельно точна: почти все покупают картошку. Однако,

каждый приходящий в овощной магазин в прошлом, покупал картошку и знает, хочет он или нет купить её ещё. Далее, менеджеры овощного магазина уже знают , что картошка пользуется спросом, и они уже так организовали выкладку товара в своём магазине, чтобы покупатели не смогли мимо неё пройти. Таким образом, чаще всего покупатель уже принял конкретное решение не покупать картошку во время этого захода в магазин, и следовательно проигнорирует рекомендацию относительно неё. Более ценна была бы рекомендация по поводу, например, замороженных овощей, о которых покупатель ещё не слышал, но которые бы ему понравились. Это пример рекомендаций, которые не прошли тест на очевидность. Очевидные рекомендации имеют 2 недостатка: 1) покупатель, заинтересованный в этих товарах, уже их приобрёл; 2) менеджерам магазина не нужны РС-системы, сообщающие им , какие товары в целом популярны. Они уже инвестировали средства в организацию своего магазина таким образом, чтобы такие товары были легко доступны покупателям.

Для анализа РС-систем нужны новые координаты измерений, учитывающие «неочевидность» рекомендаций. Один из таких параметров – степень новизны. Другой, имеющий отношение к этому, параметр – способность к неожиданным открытиям. Рекомендация о случайно возникшем объекте помогает пользователю найти интересный объект, который иначе он не смог бы обнаружить. Яркий пример разницы между новизной и способностью к неожиданным открытиям: рассмотрим РС-систему , которая просто рекомендует фильмы, поставленные самым любимым режиссёром пользователя. Если система рекомендует фильм, о котором пользователь ничего не знает – этот фильм является для пользователя новинкой, но очевидно не неожиданным приятным открытием. Пользователь скорее всего обнаружил бы этот фильм сам. С другой стороны, РС-система , которая рекомендует фильм нового режиссёра, скорее всего предоставляет неожиданно интересную рекомендацию. Рекомендации, являющиеся неожиданно интересными, также по определению являются новинками. Это различие между  способностью системы выдавать рекомендации по новым неизвестным пользователю объектам и тем из них, что могут оказаться неожиданно интересными, важно при оценке алгоритмов РС-систем, основанных на методе совместной фильтрации.

Разработать параметр, при помощи которого можно будет измерить способность системы выдавать рекомендации по неожиданно интересным объектам, очень сложно, так  как это показатель того, насколько хорошо рекомендации представляют объекты, являющиеся для пользователей как привлекательными, так и удивительными. Фактически, обычные методы измерения качества работы системы прямо противоположны этому.

### Поддержка и доверительность прогноза

Пользователи РС-систем часто сталкиваются с проблемой в определении того, как интерпретировать рекомендации по двум часто конфликтующим показателям. Первый параметр – это поддержка (support) рекомендации, т.е. насколько по мнению РС-системы пользователю понравится тот или иной объект. Такие рекомендация основываются на больших массивах данных, т.е. основываются на шаблонах, присутствующих у большинства пользователей. Второй параметр – доверительность (confidence) рекомендации, т.е. насколько сильно РС-система уверена в точности своих рекомендаций. Этот показатель основывается на вероятности выполнения рекомендуемого действия в зависимости от уже совершенных пользователем действий. Эта вероятность может быть велика, но при этом может основываться на небольшом числе случаев, т.е. на неком редком паттерне.

Чтобы помочь пользователям принять эффективное решение на основе рекомендаций, РС-системы должны помогать пользователям сориентироваться одновременно и по поддержке рекомендаций, и по доверительности. На практике применяются различные подходы. Системы e-commerce часто отказываются предоставлять рекомендации, основанные на информационных массивах, считающихся небольшими. Они хотят рекомендаций, на которые пользователи смогли бы положиться. В то же время для выявления неожиданных шаблонов и закономерностей используются рекомендации, основанные на редких случаях, но обладающие высокой условной вероятностью.

### Оценка степени достижения объективных целей

Для любой задачи должна быть разработана соответствующая метрика, которая определяет, что может считаться успешным итогом (результатом) работы системы. С точки зрения бизнеса целью может быть повышение прибыли интернет-магазина, посещаемости сайта и т.п. Если смотреть с позиции системы, то основным показателем качества её работы может быть точность. Однако с позиции пользователя, параметры качества работы системы должны устанавливаться в соответствии с их конкретными задачами.

При оценке степени достижения цели с точки зрения пользователя используются явные и неявные оценки. Основное отличие – когда у пользователей в явной форме спрашивается об их реакции на работу системы и когда ведётся наблюдение за их поведением. Первый тип оценки обычно использует методы интервьюирования и опроса. Второй тип обычно включает в себя ведение лога (протокола) пользовательского поведения, который впоследствие становится предметом различного рода анализов.

## Литература


1. J. Herlocker, J. Konstan, L. Terveen, and J. Riedl.: “Evaluating collaborative filtering recommender systems”, ACM Translations on Information Systems, Vol. 22(1), 2004.
2. Jun Wang, Arjen P. de Vries, Marcel J.T. Reinders, “Unifying Userbased and Itembased Collaborative Filtering Approaches by Similarity Fusion”.
3. Justin Basilico, Thomas Hofmann “Unifying Collaborative and Content-Based Filtering”.
4. Manolis G. Vozalis, Konstantinos G. Margaritis “Applying SVD on Generalized Item-based Filtering”.
5. David M. Pennock, Eric Horvitz, Steve Lawrence and C. Lee Giles “Collaborative Filtering by Personality Diagnosis: A Hybrid Memory- and Model-Based Approach”.
6. Prem Melville, Raymond J. Mooney, Ramadass Nagarajan “Content-Boosted Collaborative Filtering for Improved Recommendations”.
7. Anne Yun-An Chen, Dennis McLeod “Collaborative Filtering for Information Recommendation Systems”.
8. Luo Si, Rong Jin “Flexible Mixture Model for Collaborative Filtering”.
9. Lyle H. Ungar, Dean P. Foster “A Formal Statistical Approach to Collaborative Filtering”.
10. Rong Jin, Joyce Y. Chai, Luo Si “An Automatic Weighting Scheme for Collaborative Filtering”.
11. Thomas Hofmann, Jan Puzieha “Latent Class Models for Collaborative Filtering”.
12. Thomas George, Srujana Merugu “A Scalable Collaborative Filtering Framework based on Co-clustering”.

13. A. Banerjee, I. Dhillon, J. Ghosh, S. Merugu, and D. Modha “A generalized maximum entropy approach to Bregman co-clustering and matrix approximation”.
14. Gui-Rong Xue, Chenxi Lin, Qiang Yang, WenSi Xi, Hua-Jun Zeng, Yong Yu1, Zheng Chen “Scalable Collaborative Filtering Using Cluster-based Smoothing”.
15. Zan Huang, Hsinchun Chen, Daniel Zeng “Applying Associative Retrieval Techniques to Alleviate the Sparsity Problem in Collaborative Filtering”.
16. Kai Yu, Anton Schwaighofer, Volker Tresp, Wei-Ying Ma, HongJiang Zhang “Collaborative Ensemble Learning: Combining Collaborative and Content-Based Information Filtering via Hierarchical Bayes”.
17. Dmitry Y. Pavlov, David M. Pennock “A Maximum Entropy Approach To Collaborative Filtering in Dynamic, Sparse, High-Dimensional Domains”.
18. Michael Leben “Applying Item-based and User-based collaborative filtering on the Netflix data”.
19. Benjamin Marlin “Collaborative Filtering: A Machine Learning Perspective”.
20. А.А. Барсегян, М.С. Куприянов, И.И. Холод, М.Д. Тесс “Анализ данных и процессов”, СПб.: БХВ-Петербург, 2009.