Лекция. Рекомендательные системы

Вопросы

1. Характеристики рекомендательных систем.
2. Ассоциативные правила.
3. Фильтрация по содержимому.
4. Коллаборативная (совместная) фильтрация.

1. Характеристики рекомендательных систем.

Рекомендательная система — комплекс алгоритмов, программ и сервисов, задача которого предсказать, что может заинтересовать того или иного пользователя. В основе работы лежит информация о профиле человека и иные данные.

Типы рекомендательных систем:

- на основе ассоциативных правил;
- основанные на контенте (content-based);
- основанные на знаниях (knowledge-based);
- коллаборативная (совместная) фильтрация.

Рекомендательные системы на основе ассоциативных правил работают на основе простого статистического анализа совместной встречаемости различных объектов.

В рекомендательных системах, основанных на контенте те или иные объекты рекомендуются на основе знаний о них: жанр, производитель, конкретные функции и т.п. В общем, применяют любые данные, которые можно собрать.

В рекомендательных системах, основанных на знаниях объекты рекомендуются на основе знаний о какой-то предметной области.

При коллаборативной фильтрации рекомендации основаны на истории оценок как самого пользователя, так и других. Во втором случае системы рассматривают потребителей, оценки или интересы которых похожи на ваши.

Рекомендательные системы работают на двух уровнях:

- глобальные оценки; особенности и предпочтения, не меняющиеся долгое время.
- кратковременные тренды и быстрые изменения интересов во времени.

Данные собирают «явным» и/или «неявным» способами. В первом случае посетителю предлагают заполнять анкеты, проходить опросы и т.п. Второй метод предусматривает фиксирование поведения потребителя на сайте или в приложении.

Грамотно настроенный сбор информации позволяет сделать рекомендации релевантными. С их помощью сокращается время поиска нужных объектов, а также повышается вероятность совершения сопутствующих целевых действий.

2. Ассоциативные правила

Ассоциативные правила позволяют находить закономерности между связанными событиями.

Определение. $I = \{i_1, i_2, \dots, i_n\}$ - множество (набор) товаров, называемых элементами. D – множество транзакций, где транзакция T – это набор элементов из $I, T \subseteq I$. $T = (\dots, t[k], \dots)$, где $t[k] = 1$, если i_k элемент присутствует в транзакции, иначе $t[k] = 0$. Транзакция T содержит X, некоторый набор элементов из I, если $X \subset T$. Ассоциативным правилом называется импликация $X \Rightarrow Y$, где $X \subset I, Y \subset I$ и $X \cap Y = \varnothing$. Правило $X \Rightarrow Y$ имеет поддержку s (support), если s% транзакций из D, содержат $X \cup Y, supp(X \Rightarrow Y) = supp(X \cup Y)$. Достоверность правила показывает какова вероятность того, что из X следует Y. Правило $X \Rightarrow Y$ справедливо с достоверностью (confidence) c, если $c\%$ транзакций из D, содержащих X, также содержат Y, $conf(X \Rightarrow Y) = supp(X \cup Y)/supp(X)$.

Еще одна метрика — Lift. Она вычисляется следующим образом:

- вычисляется support совместной встречаемости двух продуктов;
- делится на произведение support каждого из этих продуктов.

$$lift(x_1 \cup x_2) = \text{supp}(x_1 \cup x_2)/\text{supp}(x_1) \times supp(x_2)$$

Lift показывает, насколько items зависят друг от друга.

Алгоритмы поиска ассоциативных правил предназначены для нахождения всех правил XY, причем поддержка и достоверность этих правил должны быть выше некоторых наперед определенных порогов, называемых соответственно минимальной поддержкой (minsupport) и минимальной достоверностью (minconfidence).

Задача нахождения ассоциативных правил разбивается на две подзадачи:

1. Нахождение всех наборов элементов, которые удовлетворяют порогу minsupport. Такие наборы элементов называются часто встречающимися.

2. Генерация правил из наборов элементов, найденных согласно п.1. с достоверностью, удовлетворяющей порогу minconfidence.

Один из алгоритмов для построения ассоциативных правил - **Apriori.**

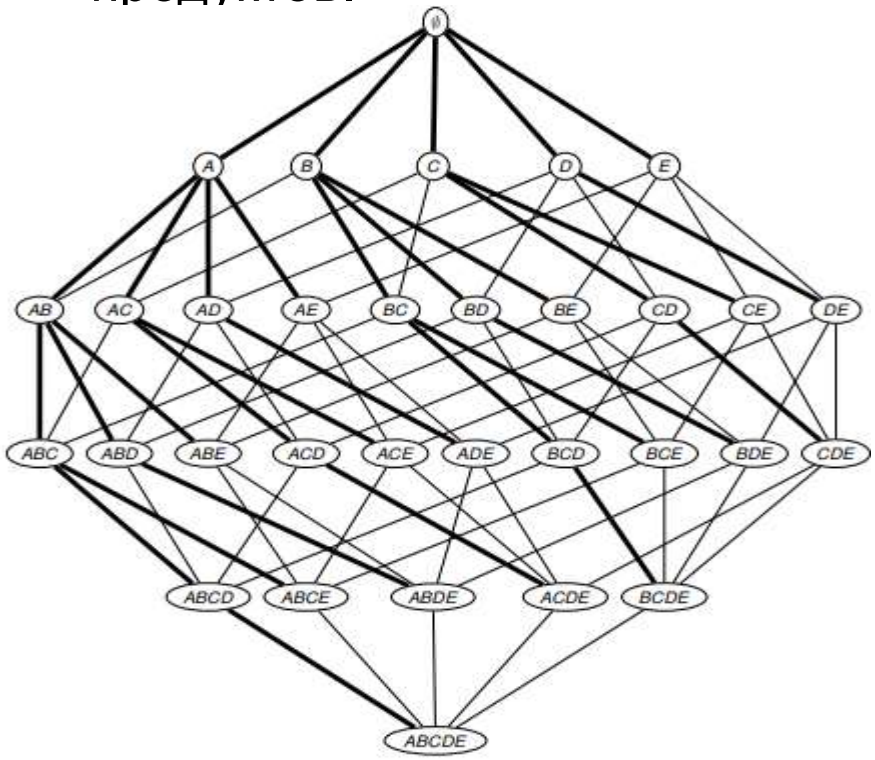
Apriori использует следующее утверждение:

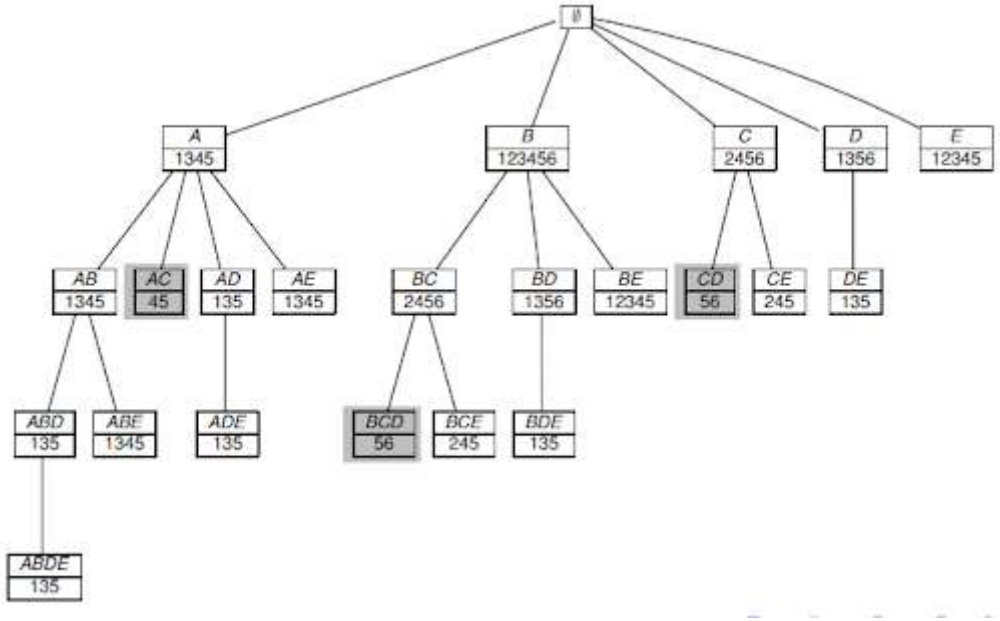
если $X \subseteq Y$, то $supp(X) \geq supp(Y)$.

Отсюда следуют 2 свойства:

- если Y встречается часто, то любое подмножество X: $X \subseteq Y$ также встречается часто
- если X встречается редко, то любое супермножество Y: $Y \supseteq X$ также встречается редко

Древовидная структура совместной встречаемости различных продуктов.





Apriori алгоритм по-уровнево проходит по префиксному дереву и рассчитывает частоту встречаемости подмножеств X в D. Таким образом:

- исключаются редкие подмножества и все их супермножества
- рассчитывается $supp(X)$для каждого подходящего кандидата X размера k на уровне k.

+ Получаем красивый граф связей между покупками продуктов.

- Можем рекомендовать только новые товары
- Не всегда можем получить требуемое число рекомендаций

Обобщенные ассоциативные правила (Generalized Association Rules)

При поиске ассоциативных правил мы предполагали, что все анализируемые элементы однородны. Однако не составит большого труда дополнить транзакцию информацией о том, в какую товарную группу входит товар и построить иерархию товаров.

Определение. Обобщенным ассоциативным правилом называется импликация $X \Rightarrow Y$, где $X \subset I$, $Y \subset I$ и $X \cap Y = \varnothing$ и где ни один из элементов, входящих в набор Y, не является предком ни одного элемента, входящего в X. Поддержка и достоверность подсчитываются так же, как и в случае ассоциативных правил.

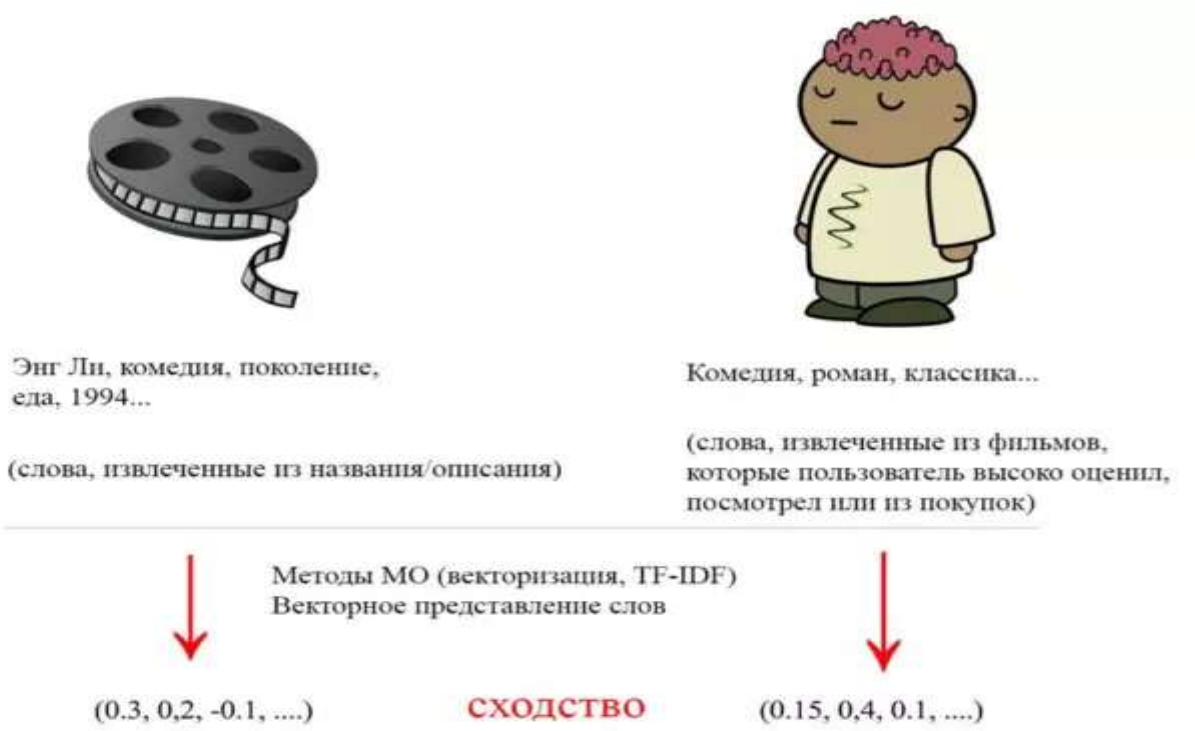
Введение дополнительной информации о группировке элементов в виде иерархии даст следующие преимущества:

- Это помогает установить ассоциативные правила не только между отдельными элементами, но и между различными уровнями иерархии (группами).

- Отдельные элементы могут иметь недостаточную поддержку, но в целом группа может удовлетворять порогу minsupport.

3. Фильтрация по содержимому

Фильтрация по содержимому (content-based filtering) выдает рекомендации на основе сходства признаков предмета (item features) и предпочтений пользователя. То есть мы можем извлечь значимые признаки из описания фильма и сопоставить их с предпочтениями пользователя.



Собираются описания объектов, которые заинтересовали пользователя, и используются в качестве входных данных для извлечения предпочтений пользователя.

```
"John","likes","to","watch","movies","Mary","likes","movies","too"

"John","also","likes","to","watch","football","games"
```

```
BoW1 = {"John":1,"likes":2,"to":1,"watch":1,"movies":2,"Mary":1,"too":1};
BoW2 = {"John":1,"also":1,"likes":1,"to":1,"watch":1,"football":1,"games":1};
```

Например, используется метод TF-IDF для пересчета значений векторе. Это снижает значение слов, которые часто встречается в описаниях фильмов.

$$\text{tf-idf}_{ij} = \text{tf}_{ij} \times \log\left(\frac{N}{\text{df}_i}\right)$$

общее число документов

частота i в док. j

номер док., содержащего i

$$\cos\theta = \frac{u \cdot v}{||u||\,||v||}$$

4. Коллаборативная (совместная) фильтрация.

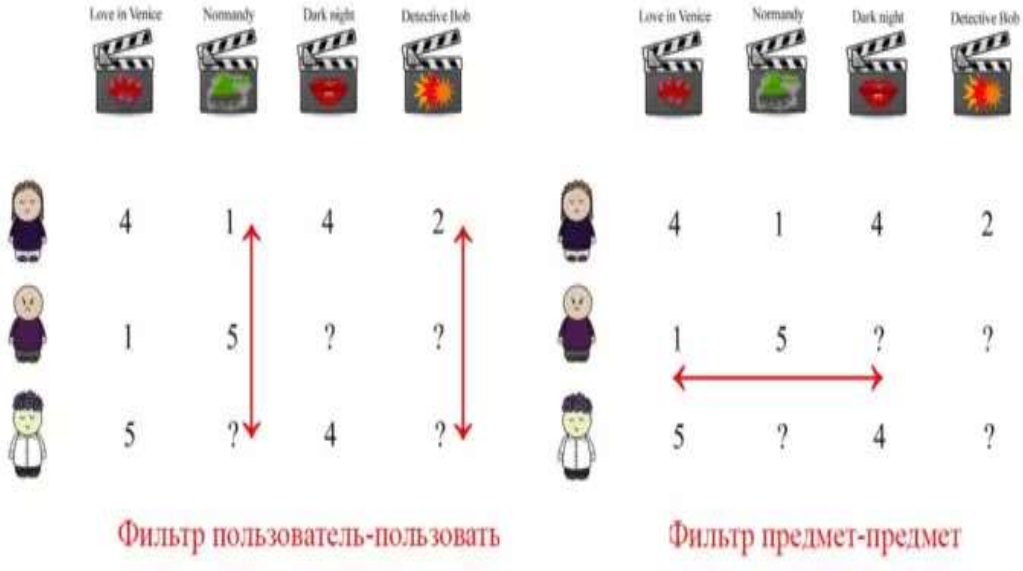
Совместная фильтрация (Пользователь-пользователь или предмет-предмет)

Фильтрация по содержимому имеет ограничение по качеству описания, предоставляемого контент-провайдером. Технически, существует ограничение на то, какие признаки могут быть извлечены из ограниченного объема контентной информации.

При совместной (коллаборативной) фильтрации концентрируются на составлении прогнозов на основе поведения пользователей, а не на извлечении признаков.

При совместной фильтрации предоставляются метки (labels), но не контентные признаки. Совместная фильтрация собирает информацию о том, как происходит взаимодействие с предметами — что вы оцениваете, как вы оцениваете, что вы просматриваете и/или что вы покупаете.

	Товар 1	Товар 2	Товар 3	Товар 4	Товар 5
Клиент 1		3		5	
Клиент 2	1		1	1	
Клиент 3	2			3	2
Клиент 4		4			5
Клиент 5	5		2	3	4



Кластеризация пользователей

Применяется алгоритм кластеризации для того, чтобы объединить людей. Выбирается условная мера схожести пользователей по их истории оценок: $sim(u, v)$.

Пользователи объединяются в группы (кластеры): $u \longmapsto F(u)$.

Оценка пользователя объекту предсказывается как средняя оценка кластера этому объекту: $\hat{r}_{ui} = \frac{1}{|F(u)|} \sum_{v \in F(u)} r_{vi}$

Проблемы алгоритма:

- Нечего рекомендовать новым/нетипичным пользователям.
- Не учитывается специфика каждого пользователя.
- Если в кластере никто не оценивал объект, то предсказание сделать не получится.

Для того, чтобы преодолеть эти сложности необходимо уйти от решения задачи кластеризации. Для этого по каждому клиенту подбирается релевантный для него товар в рамках группы клиентов, но не решается задача кластеризации, а усредняются интересы данной группы в дистанции нескольких соседей. Используя метрику дистанции между людьми, каждый человек сможет получить рекомендацию того или иного товара по рекомендациям соседних клиентов. В данном случае не все ячейки матрицы предпочтений будут заполнены и не все items получат рекомендации.

Если транспонировать матрицу предпочтений и решать ту же самую задачу не в рамках клиентов, а в рамках items, то получается более устойчивое решение. Будет большая размерность на каждый вектор items, чем размерность вектора клиентов по items. За счет этого будет больше оценок, выше статистическая значимость и модель будет более устойчива к переобучению.

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8
u_1	?	4.0	4.0	2.0	1.0	2.0	?	?
u_2	3.0	?	?	?	5.0	1.0	?	?
u_3	3.0	?	?	3.0	2.0	2.0	?	3.0
u_4	4.0	?	?	2.0	1.0	1.0	2.0	4.0
u_5	1.0	1.0	?	?	?	?	?	1.0
u_6	?	1.0	?	?	1.0	1.0	?	1.0
u_a	?	?	4.0	3.0	?	1.0	?	5.0
r_a	3.5	4.0			1.3		2.0	

$$\hat{r}_{aj} = \frac{1}{\sum_{i \in \mathcal{N}(a)} s_{ai}} \sum_{i \in \mathcal{N}(a)} s_{ai} r_{ij}$$

Преимущества

Меньше размерность матрицы расстояний

Модель более устойчива к переобучению

Можно реже обновлять

Меньше подвержены изменению предпочтений со временем

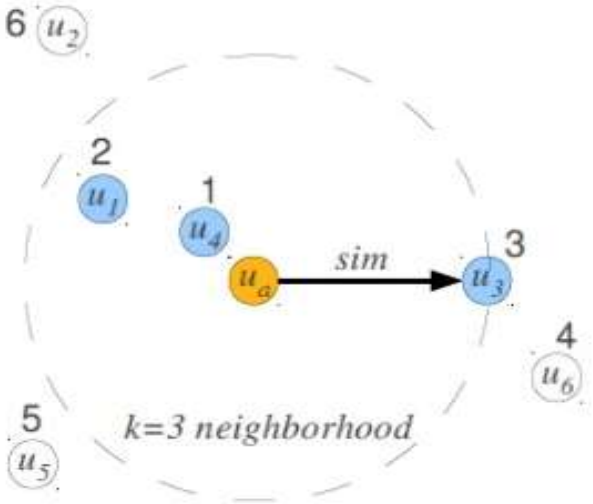
Недостатки

Проблема холодного старта

Плохие предсказания для новых/нетипичных пользователей/объектов

Тривиальность рекомендаций

Ресурсоемкость вычислений



$$\text{sim}_{\text{Cosine}}(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|}$$

Совместная фильтрация (Разложение матрицы)

Singular Value Decomposition (SVD)

В теореме о сингулярном разложении утверждается, что у любой матрицы размера $n \times m$ существует разложение в произведение трех матриц U, Σ, V^T:

$$A_{n\times m} = U_{n\times m} \times \Sigma_{n\times m} \times V^T{}_{n\times m}$$

Матрицы U и V ортогональные, а Σ — диагональная

Столбцы матриц U и V – левый и правый сингулярные векторы, а значения на диагонали Σ – соответствующие им сингулярные значения.

$AV = U\Sigma V^T V = U\Sigma = W$ – представление A в терминах ее главных компонент. Первая строка – первая главная компонента, вторая строка – вторая главная компонента и т.д.

У этих главных компонент имеется геометрическая интерпретация: направление максимального разброса данных в A, второго по величине разброса и т. д. В предположении, что данные центрированы относительно нуля, эти направления можно получить с помощью комбинации поворота и масштабирования, в этом и заключается смысл метода PCA.

$$\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 2 & 2 & 2 \\ 0 & 0 & 0 & 1 \\ 1 & 2 & 3 & 2 \\ 1 & 0 & 1 & 1 \\ 0 & 2 & 2 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \times \begin{vmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{vmatrix} \times \begin{vmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{vmatrix}.$$

Матрица в левой части выражает предпочтения людей в части фильмов (по столбцам). Правая часть – это разложение, или факторизация данной матрицы по жанрам фильмов: первая матрица количественно выражает оценку жанров, последняя ассоциирует фильмы с жанрами, а средняя дает веса каждого жанра при определении предпочтений.

Из матрицы рейтингов узнаются скрытые факторы (главные компоненты), которые пользователи используют при оценивании фильмов. Например, этими скрытыми факторами могут быть жанр фильма, год выхода, актер или актриса в фильме. Но они не определяются однозначно. Мы не знаем, правильные ли скрытые факторы определил компьютер, пока не изучим результаты вручную.

$$A_{n\times m} = U_{n\times m} \times \Sigma_{n\times m} \times {V^T}_{n\times m} = \sigma_1 {u_1}_{m\times 1} {v^T}_{1_{1\times n}} + \cdots + \sigma_r {u_r}_{m\times 1} {v^T}_{r_{1\times n}}$$

$$U\Sigma V^T = \sum_{i=1}^{r} U_i \sigma_i (V_i)^T$$

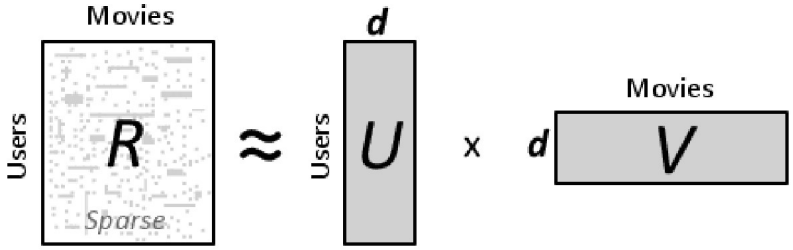
$$\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 2 & 2 & 2 \\ 0 & 0 & 0 & 1 \\ 1 & 2 & 3 & 2 \\ 1 & 0 & 1 & 1 \\ 0 & 2 & 2 & 3 \end{bmatrix} = \begin{vmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{vmatrix} + \begin{vmatrix} 0 & 0 & 0 & 0 \\ 0 & 2 & 2 & 2 \\ 0 & 0 & 0 & 0 \\ 0 & 2 & 2 & 2 \\ 0 & 0 & 0 & 0 \\ 0 & 2 & 2 & 2 \end{vmatrix} + \begin{vmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Матрицы в правой части можно интерпретировать как модели рейтингования, обусловленные жанром.

Исходная матрица раскладывается в произведение трех матриц, причем раскладывается приблизительно. Чтобы предсказать оценку пользователя:

- берется некоторый вектор (набор параметров) для данного пользователя
- берется вектор для данного фильма .

Их скалярное произведение и будет нужным предсказанием: $\widehat{r_{ui}} = \langle p_u, q_i \rangle$



Почему не достаточно SVD:

- матрица оценок полностью не известна;
- SVD-разложение не единственное

$$E_{(u,i)}(r_{ui}(\Theta) - \widehat{r_{ui}})^2 \to \min$$

$$\underbrace{\sum_{(u,i)\in\mathcal{D}} \left(\hat{r}_{ui}(\Theta) - r_{ui}\right)^2}_{\text{качество на обучающей выборке}} + \underbrace{\lambda \sum_{\theta\in\Theta} \theta^2}_{\text{регуляризация}} \to \min_{\Theta}$$