


5. Кластеризация товаров



Первое, что нужно сделать:
проверить, можно ли пренебречь
взаимным влиянием товаров!!!

(этот слайд будет 3 раза!)

Что такое кластеризация?



Кластерный анализ (Data clustering) — задача разбиения выборки объектов на непересекающиеся подмножества, называемые кластерами, так, чтобы каждый кластер состоял из схожих объектов, а объекты разных кластеров существенно отличались.

Кластеризация неформальная задача.

Потому что нет явной функции качества.

На одних и тех же объектах может быть много разных кластеризаций.

Что такое кластеризация?



Зачем нужна кластеризация?



Что мы уже умеем? Решать задачу динамического ценообразования.
Но! Для одного товара. И в статике (на локальном периоде).

А что, если продаж у одного товара мало? Скажем, продается 1-2 товар в день.

Чтобы накопить статистику придется ждать месяцы.

При этом в категории, в которой входит товар, продается 100 товаров в день.
Хочется объединить товары в кластер, чтобы можно было быстрее получать статистику и итерировать.

Зачем можно кластеризовать товары?



- Анализ цен
- Анализ аномалий
- Анализ частоты покупок
- Аналитика дистрибуции
- Анализ маркетинга
- Визуализация продаж

Для каждой задачи – своя кластеризация.

Нам нужно кластеризовать



Для динамического ценообразования нужно:

1. Похожие товары (чтобы лучше учитывать статистику)
2. Товары, на которые общий спрос (они канибализируют друг друга)
3. Якорный товар и его кластер
4. И заодно научиться выделять KVI 😊

Для каждой задачи – своя кластеризация.

Гало-эффект и каннибализация



1. При **гало-эффекте** продажи основного товара увеличивают продажи сопутствующих.

"Якорный товар" - это товар, с которым, как правило, докупают другие товары. Является триггером для покупки всей корзины.

2. При **каннибализации** товары, обладающие схожими характеристиками, замещают друг друга. Спрос одного уменьшает продажи другого.

Товары со схожими характеристиками называют **товарами-заменителя** (Субститутами).

KVI и ТПЦ



3. KVI (key value indicator) - товары, которые влияют на восприятие покупателем уровня цен в магазине.

- часто приобретаемые продукты – регулярность приобретения позволяет запомнить цену;
- брендовые продукты, продающиеся в большинстве магазинов – можно сравнить одну и ту же позицию у конкурентов;
- эластичные – покупатель чувствителен к изменению стоимости;

4. Товары первой цены (ТПЦ) – это товары, на которые установлена самая низкая цена на полке в каждой категории. Цены сравниваются относительно конкурентов и внутри магазина. В ТПЦ входят Социально значимые товары

Пример «Якоря»

Допустим есть 2 товара: сотовый телефон и аксессуары к нему.
И пусть известно, что **аксессуары к сотовому телефону докупают с вероятностью 10%.**

Вариант А

Сотовый телефон:

Цена (закупка)
10 500 р. (10 000 р.)

Чехол + пленка:

3 000 р. (500 р.)

10 заказов / +7 500 р.

Вариант Б:

Сотовый телефон:

10 000 р. (10 000 р.)

Чехол + пленка:

3 000 р. (500 р.)

Заказов больше в 5 (!) раз!

50 заказов / +12 500 р.


Гало-эффект и каннибализация



При **гало-эффекте** продажи основного товара увеличивают продажи сопутствующих.

При **каннибализации** товары, обладающие схожими характеристиками, замещают друг друга. Спрос одного уменьшает продажи другого.

KVI - товары, которые влияют на восприятие покупателем уровня цен в магазине.



Первое, что нужно сделать:
проверить, можно ли пренебречь
взаимным влиянием товаров!!!



Похожие товары

Что мы знаем о товаре?



У товара, как правило есть:

- Иерархия (категории)
- Свойства товара
 - Характеристики
 - Фото
 - Описание
 - Цена 😊
- История покупок

Что мы знаем о товаре?



Если категоризация (иерархия) хорошая ее можно взять за кластеризацию.

На практике так редко бывает. 😊
Почему?

Пример.

Категория ноутбуки содержит 1000+ товаров. И не содержит подкаталогов. Какие-то продаются часто. Какие-то нет.

Что делать? 😊

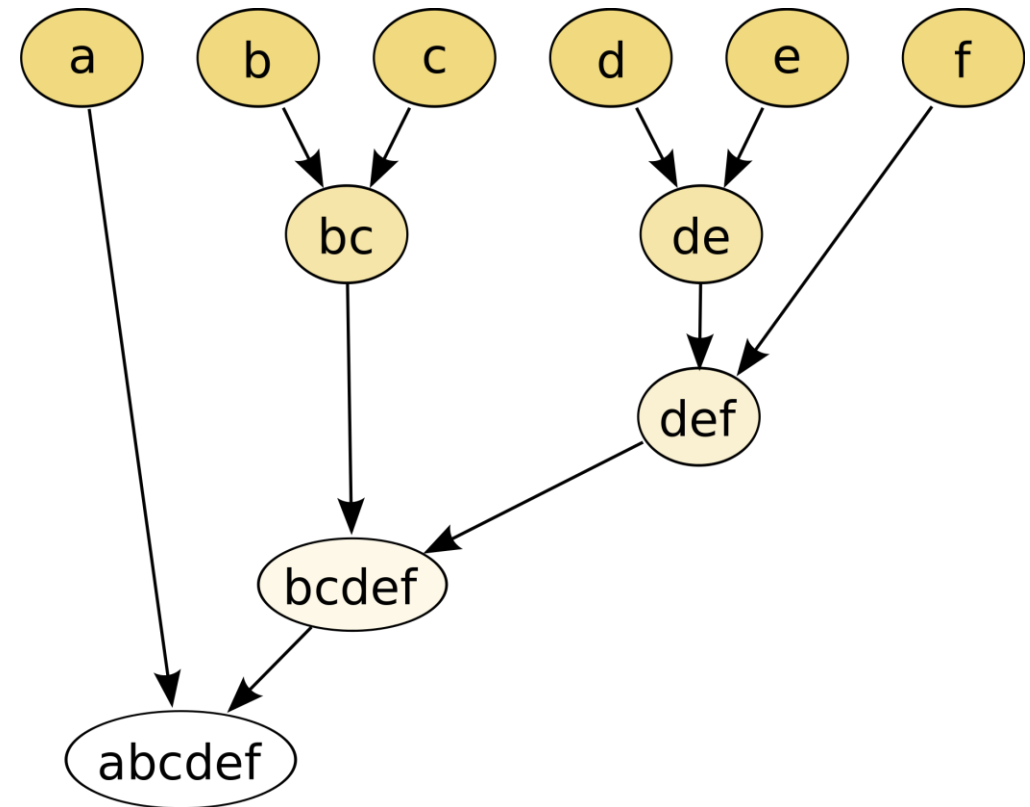
Пример каталога

Одежда (100000 покупок в неделю)

- Футболки (4000)
 - Белые футболки (1000)
 - Футболка белая из Иваново (100)
 - Футболка белая из Самары (100)
 - Черные футболки (120)
 - Цветные футболки (130)
 - Красные футболки (1)
 - Оранжевые футболки (1)
- Не кластер, потому что общий спрос (каннибализация)
- Не кластер, потому что мало разнородные товары
- Не кластер, потому что мало продаж

Иерархическая кластеризация

Иерархическая кластеризация— совокупность алгоритмов упорядочивания данных, направленных на создание иерархии (дерева) вложенных кластеров.



Иерархическая кластеризация



Классы методов иерархической кластеризации:

- **Агломеративные методы** (англ. agglomerative): новые кластеры создаются путем объединения более мелких кластеров и, таким образом, дерево создается от листьев к стволу;
- **Дивизионные методы** (англ. divisive): новые кластеры создаются путем деления более крупных кластеров на более мелкие и, таким образом, дерево создается от ствола к листьям.

Алгоритмы кластеризации



Агломеративный (снизу вверх) алгоритм

Агломеративная кластеризация начинается с n кластеров, где n — число наблюдений: предполагается, что каждое из них представляет собой отдельный кластер. Затем алгоритм пытается найти и сгруппировать наиболее схожие между собой точки данных — так начинается формирование кластеров.

Алгоритмы кластеризации



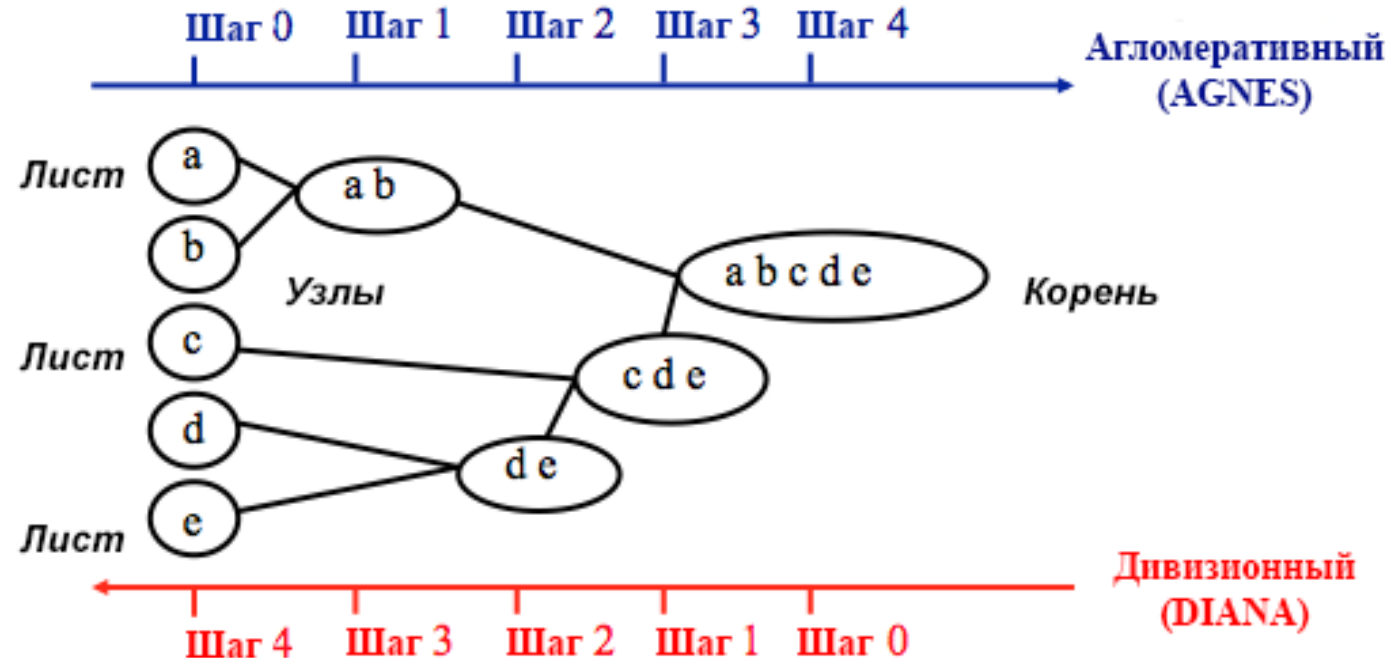
Дивизионный (сверху вниз) алгоритм

Дивизионная кластеризация выполняется противоположным образом — исходно предполагается, что все n точек данных, которые у нас есть, представляют собой один большой кластер, а далее наименее схожие из них разделяются на отдельные группы.

Алгоритмы кластеризации

Иерархическая кластеризация:

- агломеративный (снизу вверх) алгоритм
- дивизионный (сверху вниз) алгоритм



Иерархическая кластеризация



- https://ru.wikipedia.org/wiki/%D0%98%D0%B5%D1%80%D0%B0%D1%80%D1%85%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D0%BA%D0%BB%D0%B0%D1%81%D1%82%D0%B5%D1%80%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F
- <http://www.machinelearning.ru/wiki/images/c/ca/Voron-ML-Clustering.pdf>
- <https://habr.com/ru/post/101338/>
- ...

Какая лучше?



По-большому счету нет разницы! 😊

Лишь бы была хорошая!

А как понять, что она хорошая?

Оценка качества кластеризации



В целом, при создании кластеров вы заинтересованы в получении четко выраженных групп так, чтобы расстояние между такими точками внутри кластера было минимальным, а расстояние между группами (отделимость) — максимально возможным.

На практике можно просто посмотреть кластеризацию на адекватность, выбрав несколько кластеров.

Также их можно (и нужно) показать товароведам!

Оценка качества кластеризации



Оценку качества мы сможем понять, только применяя ее к задаче ДЦ.

И получая ответ от среды (продажи, выручку), перестраивать кластеризацию.

Векторные представления товаров



Embedding (эмбединг)— это способ представлять объекты, когда каждый объект превращается в вектор фиксированной длины, и близким объектам соответствуют близкие векторы. Практически всем известным моделям требуется, чтобы данные на входе были фиксированной длины, и набор векторов — простой способ привести их к такому виду.

Эмбединг можно получать из чего угодно:

Из текстов — `word2vec`.

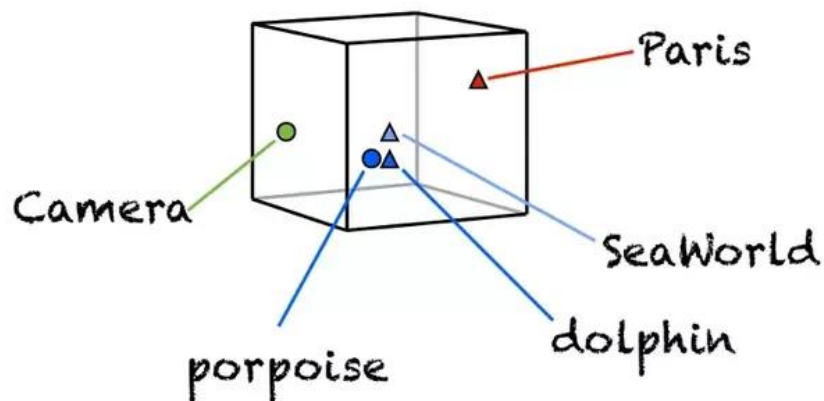
Фото — последний слой нейронной сети для задачи классификации.

Из характеристик — например **РСА** (метод главных компонент).

Embedding (векторное представление)

▪

	dim-0	dim-1	dim-2	dim-3	dim-4	...	dim-45	dim-46	dim-47	dim-48	dim-49
title											
War and Peace	-0.279165	-0.107367	0.114153	0.143709	-0.141921	...	-0.067178	0.230711	-0.230550	0.199285	-0.099167
Anna Karenina	-0.248443	-0.000578	0.150472	0.151845	0.000908	...	-0.141615	0.178011	-0.230794	0.042102	-0.189196
The Hitchhiker's Guide to the Galaxy (novel)	-0.190761	-0.060406	0.115548	-0.249868	-0.120824	...	-0.038944	0.084992	-0.047035	-0.054157	-0.209883



Построение векторного описания товара



Есть два принципиально разных **способа построения векторного описания товара**:

- **ИСПОЛЬЗОВАТЬ КОНТЕНТ** — сверточные нейронные сети для извлечения признаков из фотографий, рекуррентные сети или мешок слов для анализа текстового описания;
- использование данных о **взаимодействиях пользователей с товаром**: какие товары и как часто смотрят/добавляют в корзину вместе с данным.

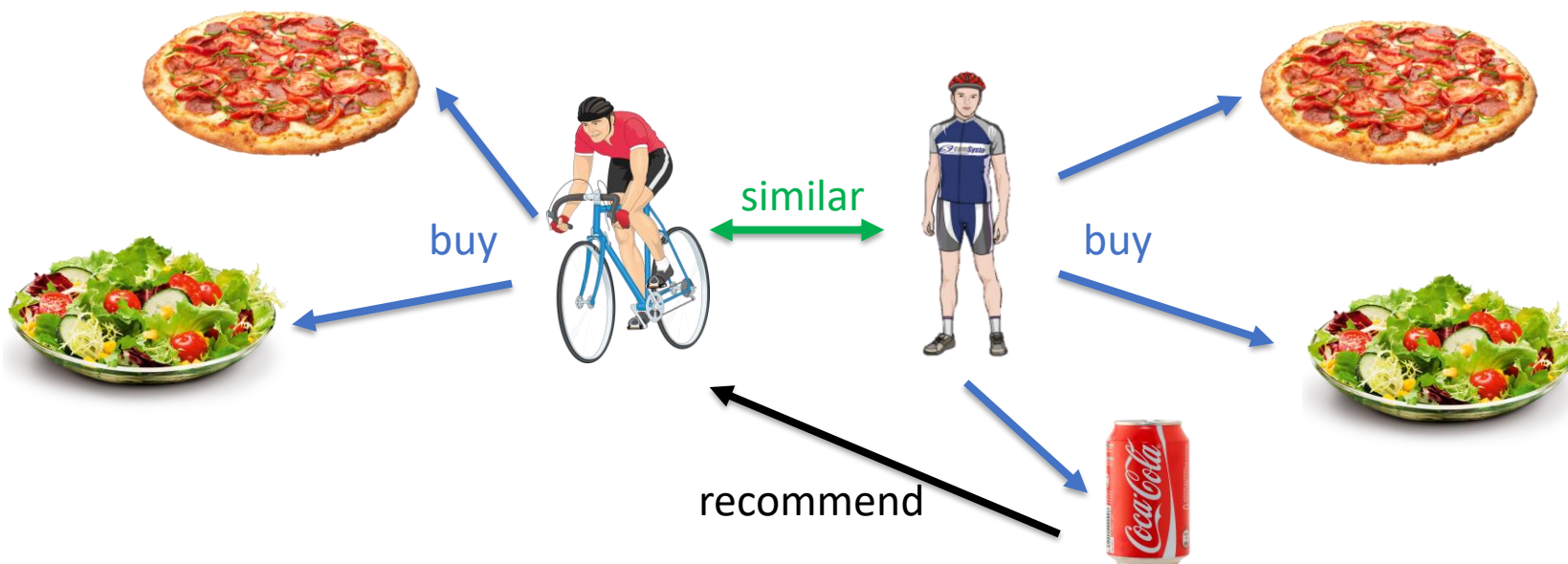


Немного о связи
с рекомендательными системам

Рекомендательные системы

Такие системы значительно упрощают поиск релевантных продуктов и обогащают опыт пользователя.

Основная идея: для пользователя найти похожих пользователей уже с историей покупок. И рекомендовать примерно тоже самое.

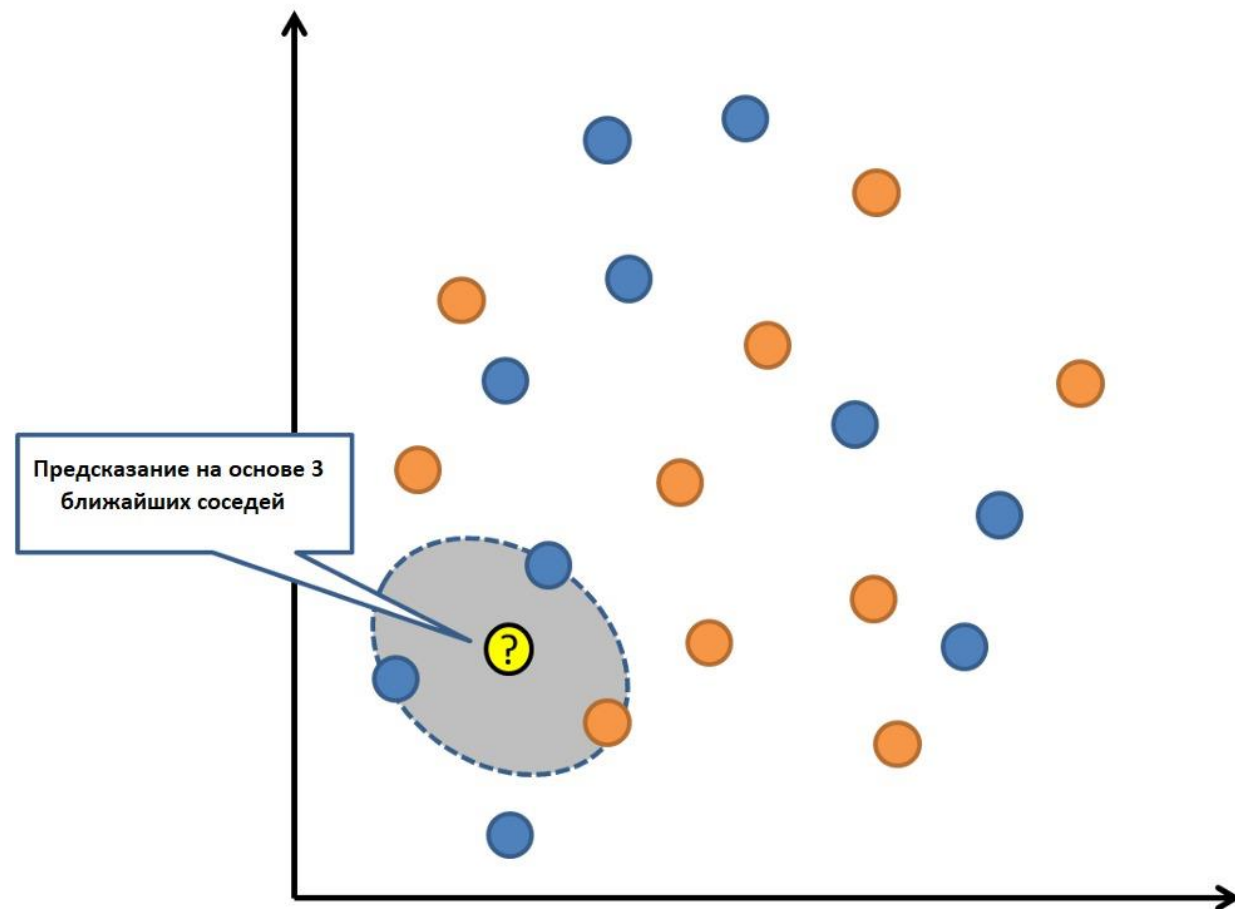


kNN

kNN (k Nearest Neighbor или k Ближайших Соседей)

Метод ближайших соседей:

Давайте введём расстояние между пользователями и будем рекомендовать то же, что нравится соседям.



User-based kNN

	1+1	Три мушкетер а	12 стульев	Легенда №17
Алексей	10	9	1	7
Борис		9	2	
Вова	1		6	
Коля	3		4	10
Петя		1		
Юля			3	6

User-based kNN



	1+1	Три мушкетер а	12 стульев	Легенда №17
Алексей	10	9	1	7
Борис		9	2	
Вова	1		6	
Коля	3		4	10
Петя		1		
Юля			3	6

User-based kNN

	1+1	Три мушкетер а	12 стульев	Легенда №17
Алексей	10	9	1	7
Борис		9	2	?
Вова	1		6	
Коля	3		4	10
Петя		1		
Юля			3	6

User-based kNN

	1+1	Три мушкетер а	12 стульев	Легенда №17
Алексей	10	9	1	7
Борис		9	2	?
Вова	1		6	
Коля	3		4	10
Петя		1		
Юля			3	6

2 ближайших
соседа по
предпочтениям

Следовательно, оценка Бориса
на фильм «Легенда №17»
будет где-то 6,5

Item-based kNN

	1+1	Три мушкетер а	12 стульев	Легенда №17
Алексей	10	9	1	7
Борис		9	2	?
Вова	1		6	
Коля	3		4	10
Петя		1		
Юля			3	6

Аналогично можно искать близость и по столбцам (по фильмам/товарам)

Переходим к товара

j

	Батончик Snickers	Мороженое Baskin Robbins	Горький шоколад	Печенье Oreo
Чек1	10	9	1	7
Чек2		9	2	
Чек3	1		6	
Чек4	3	0	4	10
Чек5		1		
Чек6			3	6

i

Можно дополнительно агрегировать чеки по пользователям.

Точно известно, что здесь 0.

Как оценить близость соседей?

Ключевым в алгоритме kNN – является расстояние (близость). От того как ее задать зависит результат.

Примеры, расстояний:

- 1) Число совпавших/совместых оценок (покупок)
- 2) Корреляция Пирсона
- 3) Косинусовое расстояние



Корреляция Пирсона

Корреляция Пирсона — классический коэффициент, который вполне применим и при сравнении векторов.

Основной его минус — когда пересечение по оценкам (покупкам) низкое, корреляция может быть высокой просто случайно.

$$\rho = \frac{\sum_i (\bar{x}_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}}$$

Косинусное расстояние

Основная идея, на которой базируется расчет косинусного расстояния, заключается в том, что строку из символов можно преобразовать в числовой вектор. Если проделать эту процедуру с двумя сравниваемыми строками, то меру их сходства можно оценить через косинус между двумя числовыми векторами.

$$\text{similarity} = \cos(\theta) = \frac{XY}{\|X\| \|Y\|} = \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2 \sum_{i=1}^n y_i^2}}$$

Из курса школьной математики известно, что если угол между векторами равен 0 (то есть векторы полностью совпадают), то косинус равен 1.

Матричная факторизация

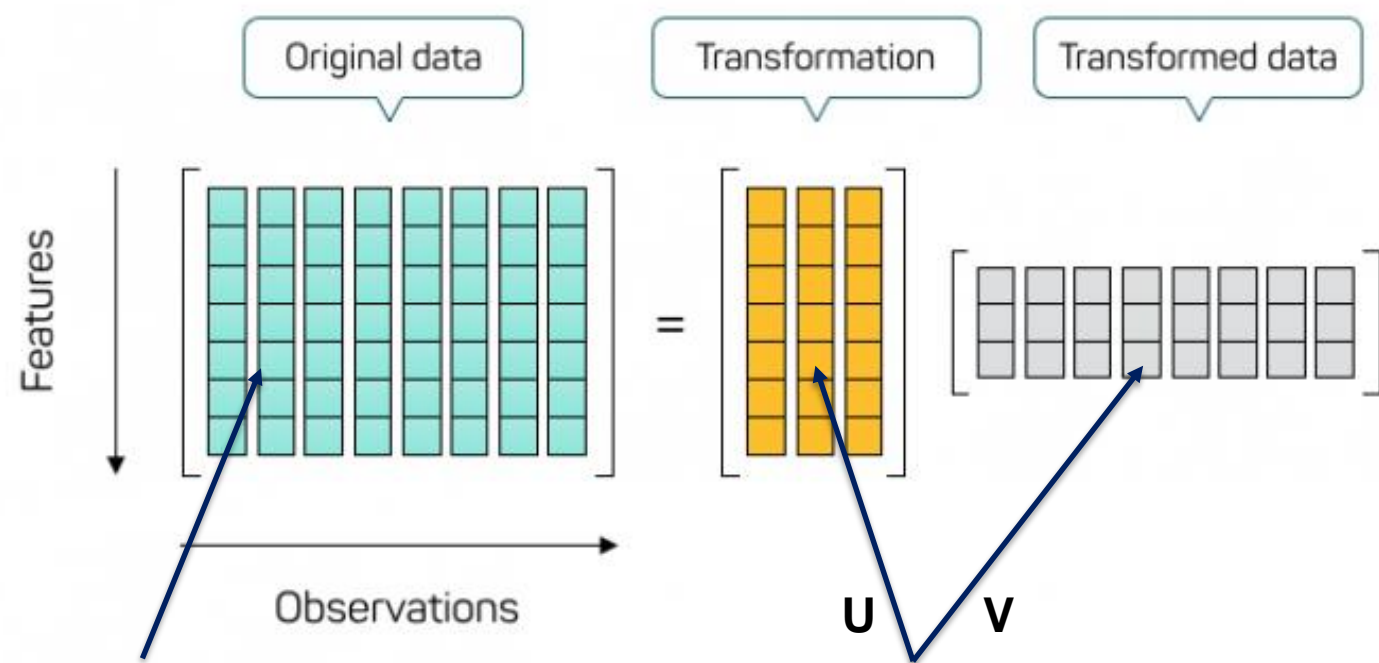
Факторизация — это операция разложения объекта (число, матрица) на его простые составляющие.

Способы разложения матрицы:

- PCA (Principal component analysis — метод главных компонент)
- SVD (Singular value decomposition — сингулярное разложение)
- NNMF/NMF (Non-negative matrix factorization- неотрицательное разложение матрицы)

Матричная факторизация

Идея факторизации матрицы:

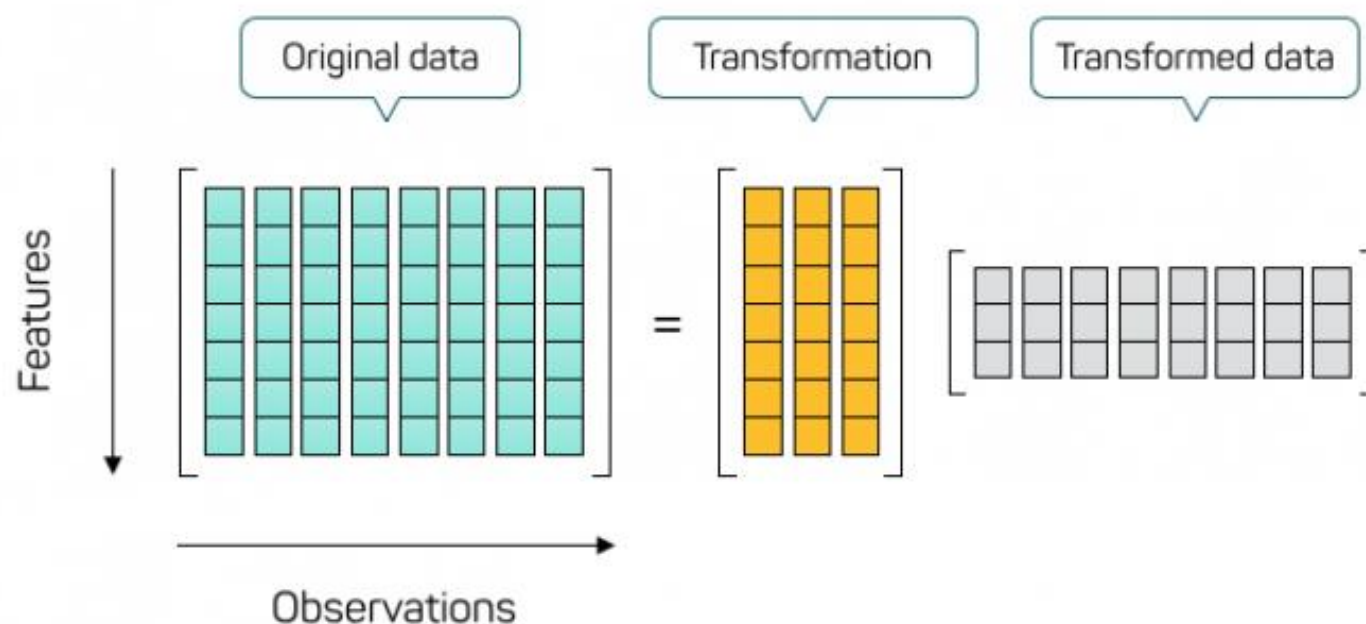


Исходная матрица (X)

Две матрицы, на которые мы
хотим разложить исходную

Матричная факторизация

Идея факторизации матрицы:



Общий вид разложения:

$$X(m \times n) = U(m \times k) \cdot V(k \times n), \text{ где } k \text{ — количество компонент.}$$

Алгоритм SVD

SVD (Singular Value Decomposition), переводится как сингулярное разложение матрицы.

- Теорема о сингулярном разложении:
у любой матрицы \mathbf{A} размера $n \times m$ существует разложение в произведение трех матриц: U , Σ и V^T :
- Матрицы U и V ортогональные, а Σ — диагональная (хотя и не квадратная).

$$\underset{n \times m}{\mathbf{A}} = \underset{n \times n}{\mathbf{U}} \times \underset{n \times m}{\mathbf{\Sigma}} \times \underset{m \times m}{\mathbf{V}^T}$$

$$\mathbf{U}\mathbf{U}^T = \mathbf{I}_n, \quad \mathbf{V}\mathbf{V}^T = \mathbf{I}_m,$$
$$\mathbf{\Sigma} = \text{diag}(\lambda_1, \dots, \lambda_{\min(n,m)}), \quad \lambda_1 \geq \dots \geq \lambda_{\min(n,m)} \geq 0$$

Алгоритм SVD

SVD (Singular Value Decomposition), переводится как сингулярное разложение матрицы.

- Помимо обычного разложения бывает еще усеченное, когда из лямбд, остаются только первые d чисел, а остальные мы полагаем равными нулю.
- Это равносильно тому, что у матриц U и V мы оставляем только первые d столбцов, а матрицу Σ обрезаем до квадратной $d \times d$.

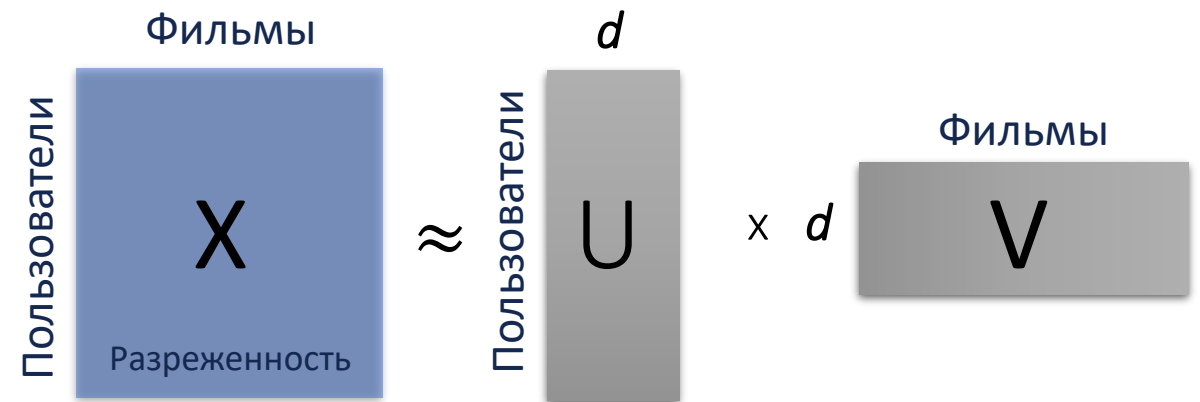
$$\lambda_{d+1}, \dots, \lambda_{\min(n,m)} := 0$$

$$\mathbf{A}'_{n \times m} = \mathbf{U}'_{n \times d} \times \mathbf{\Sigma}'_{d \times d} \times \mathbf{V}'^T_{d \times m}$$

SVD для рекомендаций

Чтобы предсказать оценку пользователя U для фильма V , мы берем некоторый вектор p_u (набор параметров) для данного пользователя и вектор для данного фильма q_i .

Их скалярное произведение и будет нужным нам предсказанием:
 $\hat{x}_{ij} = \langle u_i, v_j \rangle$



$$\hat{x}_{ui} = \langle u_i, v_j \rangle$$

Обобщение SVD

Саймон Фанк в статье в блоге, описывающей его решение Netflix Prize, предложил использовать более общий вид разложения:

$$\hat{x}_{ij} = \mu + b_i + b_j + q_j^T p_i$$

$$\hat{x}_{ij} = q_j^T p_i$$

Особенности SVD



SVD хорошо работает с рейтингами.

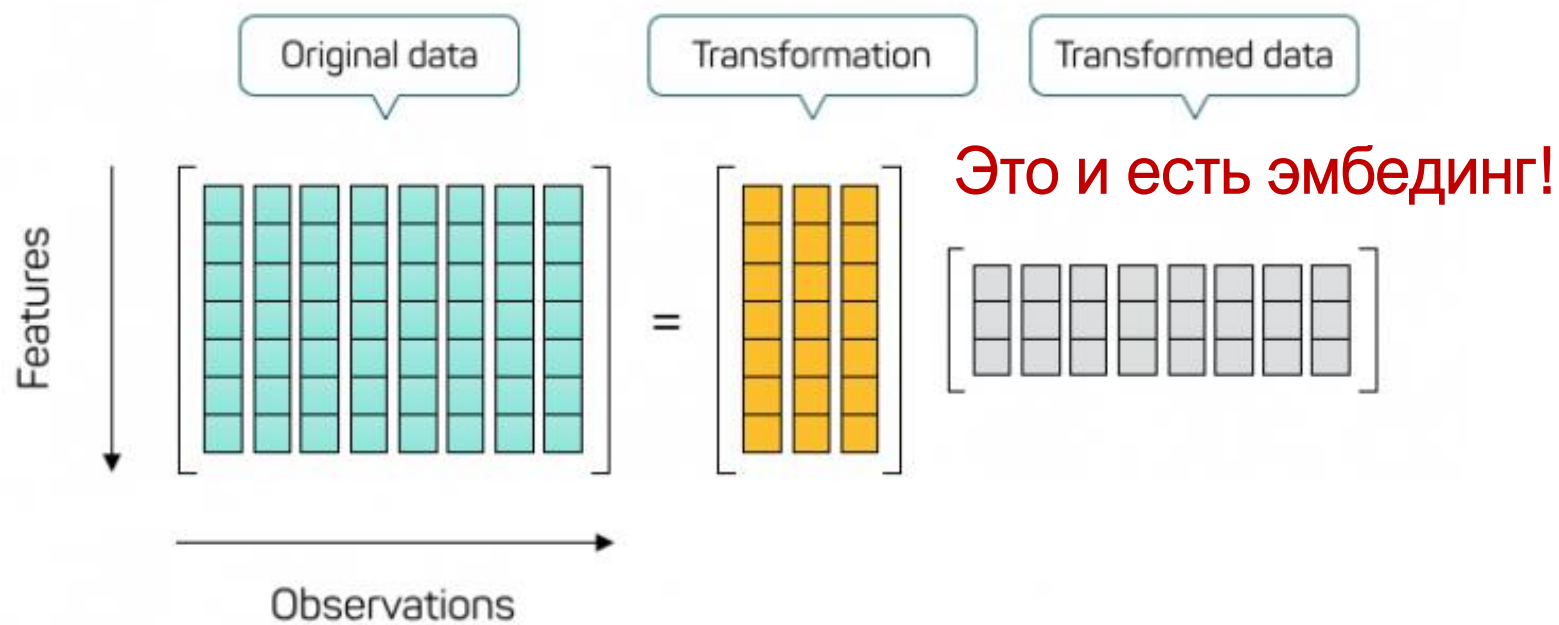
Но если матрица задана бинарными значениями (0,1), как например в случае, когда матрица заполняется покупками, SVD, как правило, показывает плохие результаты. ☹

NMF – лучше подходит для бинарных матриц.

Все тоже самое, что в SVD только обе матрицы >0 .

https://en.wikipedia.org/wiki/Non-negative_matrix_factorization

Матричная факторизация



Итого



Что есть:

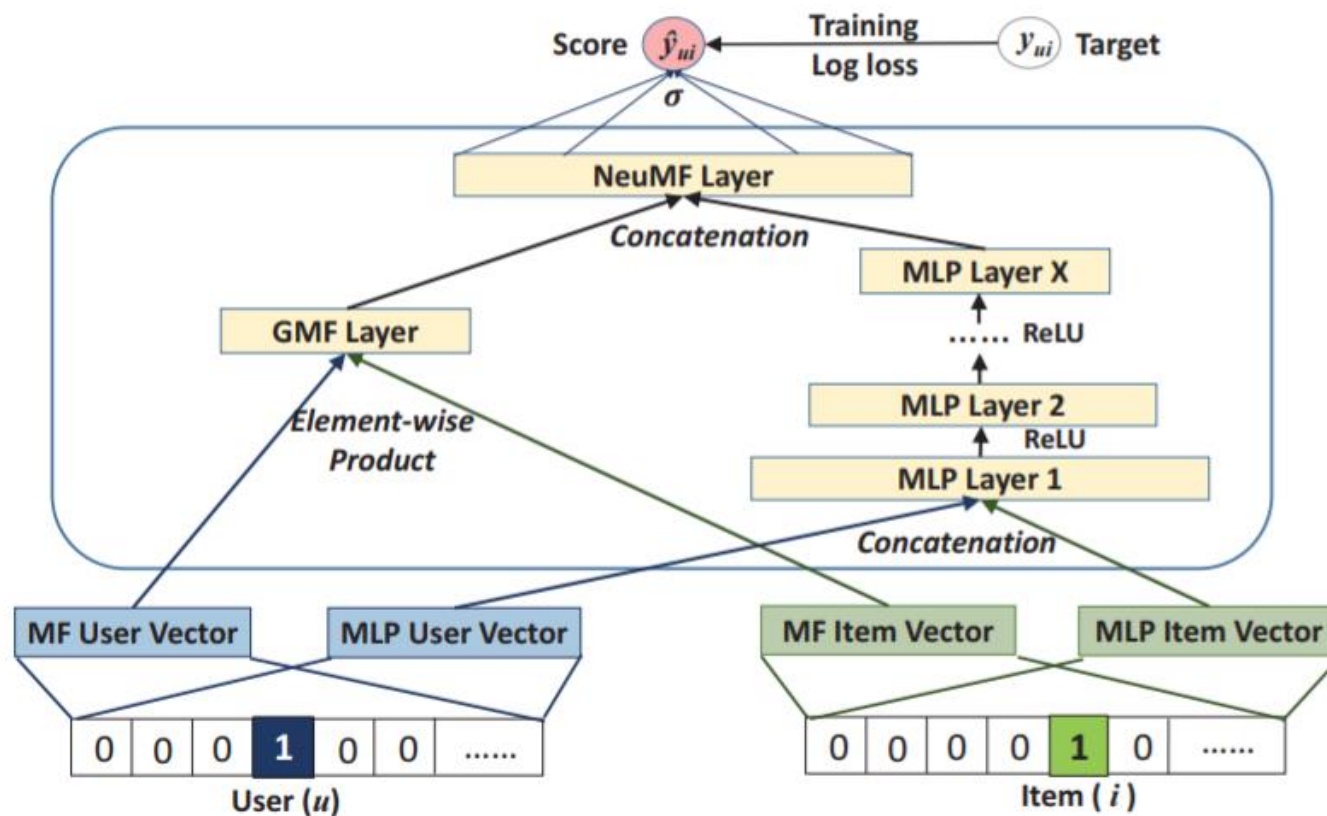
1. Есть эмбединги из свойств товара
2. Есть эмбединги из чеков (взаимодействий)

Что можно сделать:

Строить кластеризации в этих пространствах.

Дополнительный ТЮНИНГ

Neural Collaborative Filtering



Дополнительный тюнинг



Можно ставить все эмбединги «рядом» и объединять в общую архитектуру сложной нейронной сети, чтобы повысить качество представлений.

На самом деле хорошие эмбединги годятся для всего:

- Рекомендация контента
- Кластеризация по продажам

По сути мы получаем такое представление (и такое пространство) товаров, где чем ближе товары, тем они больше похожи между собой.



Канибализация

Товары-заменители (субституты)

Товары-заменители — это товары, выполняющие ту же функцию, для той же группы потребителей, но основанные на другой технологии.



Товары-заменители (субституты)

Товары-заменители — это товары, выполняющие ту же функцию, для той же группы потребителей, но основанные на другой технологии.

Эти товары создают перманентную угрозу, поскольку замещение всегда возможно. Данная опасность может возрасти, например, в результате технологических достижений, изменяющих отношение качество/цена заменителя по сравнению с существующим на рынке товаром.

Товары-заменители (субституты)



Фактически, цены на товары-заменители определяют потолок цен, которые могут назначить фирмы, действующие на рынке товара.

Чем привлекательнее для пользователей товар-заменитель, тем более ограничены возможности повышения цен на рынке товара.

Угроза товаров-заменителей

Угроза товаров-заменителей тем реальнее, чем больше:

- количество эффективных заменителей производимого товара
- объем производства товаров-заменителей
- разница в ценах между изделием-оригиналом и товарами-заменителями в пользу последних

Каннибализация



Ошибка в прогнозировании спроса может влиять на каннибализацию спроса.

Если товар А закончился, то продажи вырастут на товар В.

В таком случае нужно объединять товары А и В в общий кластер по спросу.

Как кластеризовать товары



Цель кластеризации товаров - создать кластеры такие, что:

- В каждом кластере находятся товары, на которых есть общий спрос
- Иерархические кластеры (**по спросу**)
- Для каждого кластера нижнего уровня – «работает статистика» (это мы уже сделали в предыдущем пункте)

Как проверить на каннибализацию



Проверка гипотезы осуществляется следующими подходами:

- Совместная/несовместная покупка
- «Близость» чеков с разными товарами

Как проверить на каннибализацию

Решение «в лоб»:

Перебираем по всем парам товаров (x_l, x_m) все чеки (r_l, r_m) , где присутствовали товары.

Далее для $r_{l,k}$ находим

$$\min(\text{sim}(r_{l,k}, r_{m,*}))$$

для всех k чеков с l -ым товаром.

Где sim – косинусная мера, например.

Далее считаем какой-то квантиль для набора $\min \text{sim}(r_{l,*}, r_{m,*})$, если значение близко к 0 – считаем, что товары каннибализирует.

Как проверить на каннибализацию

Хорошие новости:

1) работает транзитивность.

$$x_l \sim x_m \ \& \ x_l \sim x_w \Rightarrow x_m \sim x_w$$

2) Можно откинуть одинарные чеки, чеки в которых очень много покупок и т.д.

Как проверить на канибализацию



Разумеется, мы про товары знаем несколько больше.

Поэтому можно изначально строить смысловую иерархию (кластеры). [см. предыдущий пункт]

И проверять «в лоб», внутри кластеров на нижних уровнях.

Вряд ли (если у нас хорошие эмбединги):
товары, которые далеко друг от друга
канибализируют друг друга.

Вопросы



Как оценить качества?

- На исторических данных
- С новыми товарами

Можно ли перейти к обучению с учителем?



Гало-эффект

Как учесть гало-эффект?

Как его описать с математической точки зрения?

Пусть $Q(x_l, x_m)$ - вектор спроса (продаж).

Если он обладает таким свойством:

$$Q(x_l, x_m) = [Q_l(x_l), Q_l(x_l)z + Q_m(x_m)]$$

То x_l является «якорем» для x_m .

Для простоты можно считать, что результат спроса на l -ый товар – это добавка перед спросом на m -ый товар. С некоторым коэф. $z \neq 0$.

СМОТРИМ НА ОБЩИЕ ЧЕКИ.

Часто на практике: $Q_m(x_m) = 0$

Как учесть гало-эффект?



В самом простом случае можно построить корреляционную матрицу. Это работает когда $Q_m(x_m) = 0$.

Как учесть гало-эффект?



Вывод ассоциативных правил:

На основе имеющихся данных нужно найти закономерности между покупками.

Много ссылок:

https://en.wikipedia.org/wiki/Apriori_algorithm

<https://loginom.ru/blog/apriori>

<https://github.com/asaini/Apriori>

<https://habr.com/ru/company/ods/blog/353502/>

<https://www.softwaretestinghelp.com/fp-growth-algorithm-data-mining/>

Вывод ассоциативных правил

ID	Items
1	{Bread, Milk}
2	{Bread, Diapers , Beer , Eggs}
3	{Milk, Diapers , Beer , Cola}
4	{Bread, Milk, Diapers , Beer }
5	{Bread, Milk, Diapers, Cola}
...	...

market
basket
transactions

{Diapers, Beer}

Example of a frequent itemset

{Diapers} → {Beer}

Example of an association rule

Поиск ассоциативных правил



APRIORI

<https://share.streamlit.io/asaini/apriori/python3>

<https://github.com/asaini/Apriori>

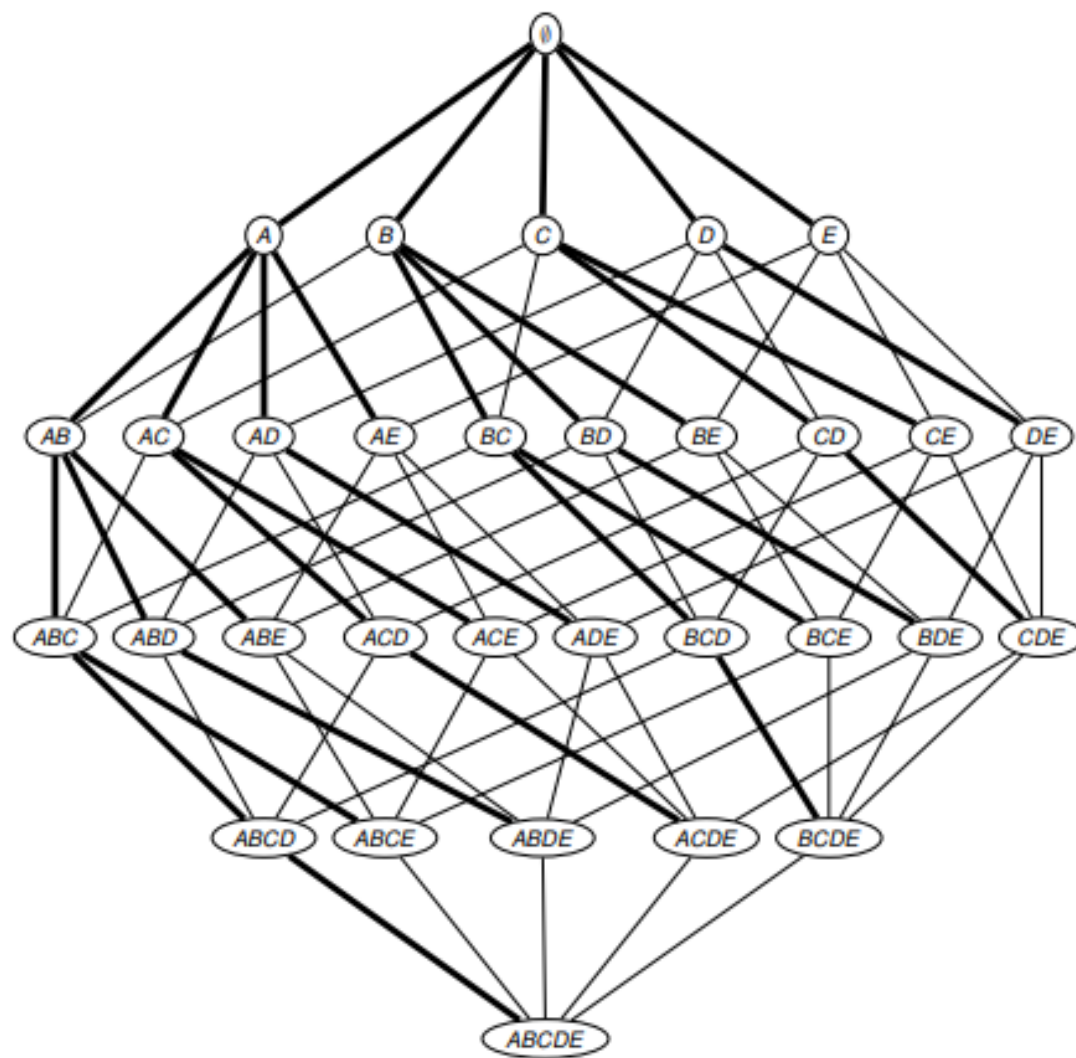
ECLAT Algorithm

<https://github.com/jeffrichardchemistry/pyECLAT>

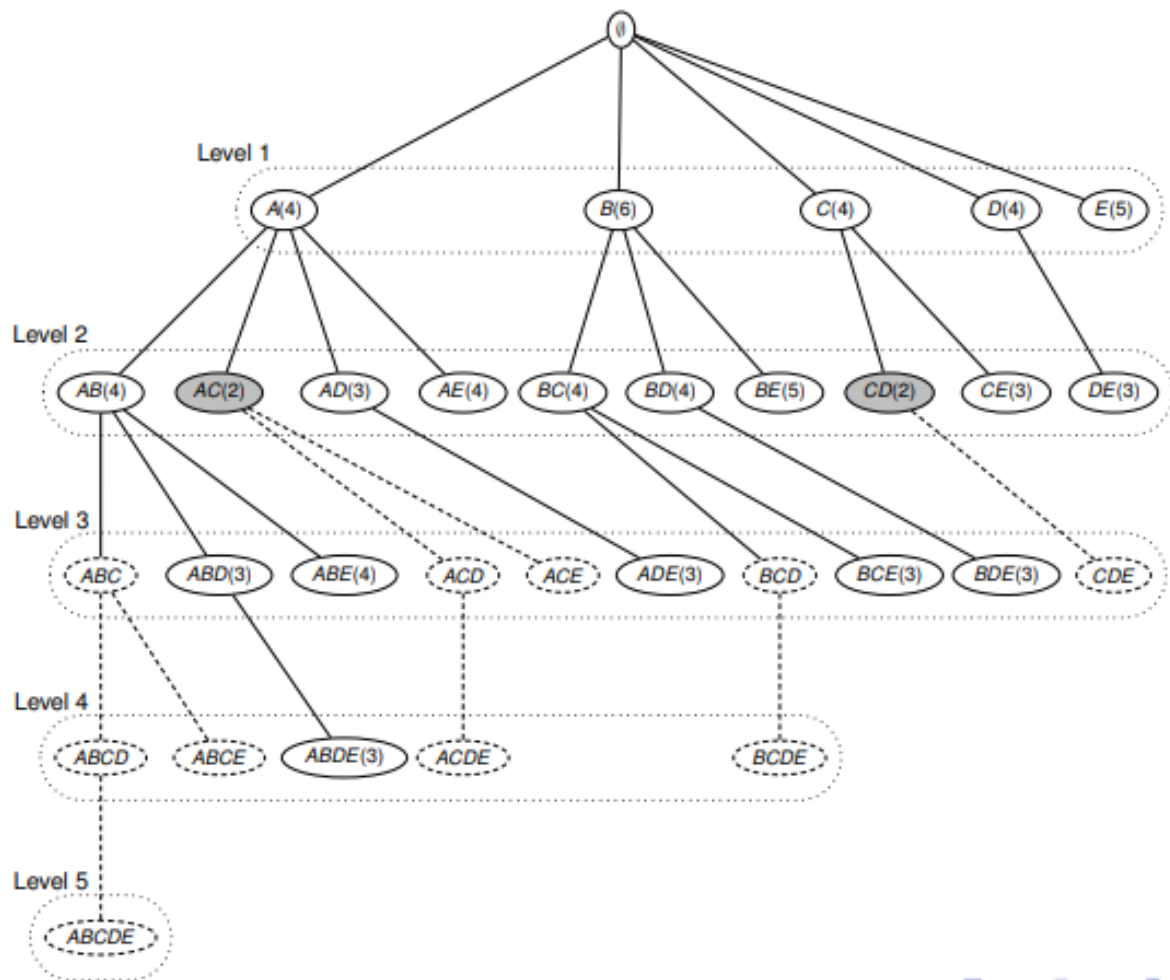
FP-Growth Algorithm

<https://github.com/enaeseth/python-fp-growth>

Общая идея всех алгоритмов



Общая идея всех алгоритмов

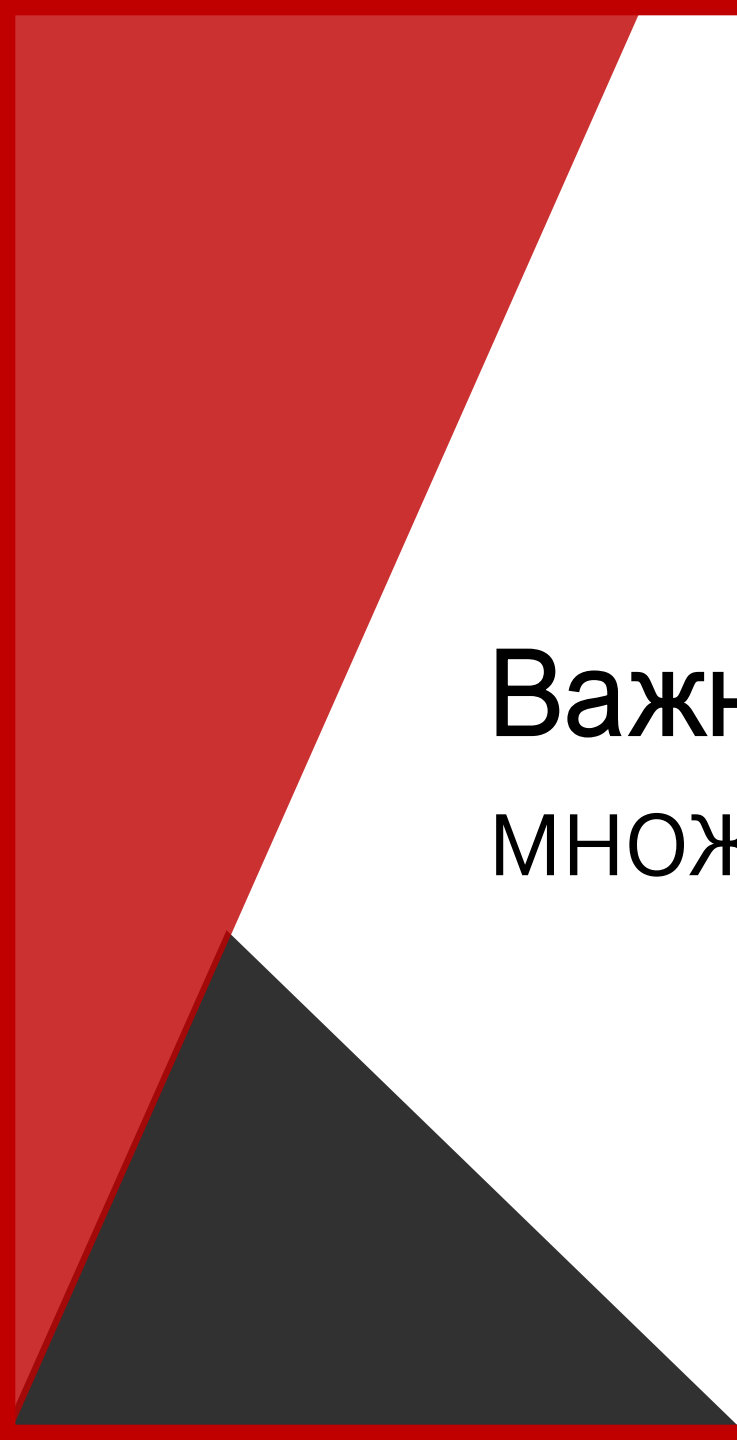


Отложенные гало-эффект?



Также есть отложенные эффекты:

- Алкоголь отрицательно коррелирует со спросом на спортивные товары через 2-3 недели
- Алкоголь положительно коррелирует с продажами «рассола» через несколько дней.



Важно: делать поправку на
множественную проверку гипотез



KVI

Как учесть KVI?



Аналогично, как и гало-эффект, только он действует *почти на все товары*, или является коэффициентом перед совокупным спросом.

Можно просто строить корреляцию между спросом и общими продажам. На практике существует некоторый коэффициент запаздывания 3-7 дней.

Как учесть KVI?

$$Q(x_{kvi}, x_m) = [Q_{kvi}(x_{kvi}), Q_{kvi}(x_{kvi})z + Q_m(x_m)]$$

Если найдена такая пара, это не означает, что x_{kvi} .

KVI «тащит» не конкретные продажи, а покупателей (т.е. увеличивает спрос).

Как учесть KVI?

Несколько идей:

1)

$$\text{corr}(Q(x_{kvi}), Q'(x)) \approx 1$$

Где, Q' - это продажи в будущем.

2) $Q(x_{kvi})$ - сильно выше среднего (т.е. kvi товары часто покупают).

3) Товары, которые покупают с x_{kvi} разнообразные. То есть $D[r(x_{kvi})] > d$

Как учесть KVI?



Также можно спросить у экспертов.

Они помогут сузить круг товаров для поиска KVI и других кросс-эффектов.

Итого



- Есть кластеризация товаров по близости
- Найдены канибализирующие товары
- Найдены якоря
- Найдены KVI

Что дальше?

Один и тот же товар может находиться в разных кластерах.
Он может быть одновременно и якорем, и товаром, который канибализирует другой.

«Якорь» для игрушек (1)

Мягкая игрушка «Медведь» 16 см

Мягкая игрушка «Слон» 16 см

Мягкая игрушка «Медведь» 30 см


Мягкая игрушка «Слон» 30 см

«Якорь» для игрушек (2)

Вопросы



- Как это учитывать при построении многомерных алгоритмов?
- Какие кластера более приоритетны?
- Как учесть якорные товары?



Первое, что нужно сделать:
проверить, можно ли пренебречь
взаимным влиянием товаров!!!