

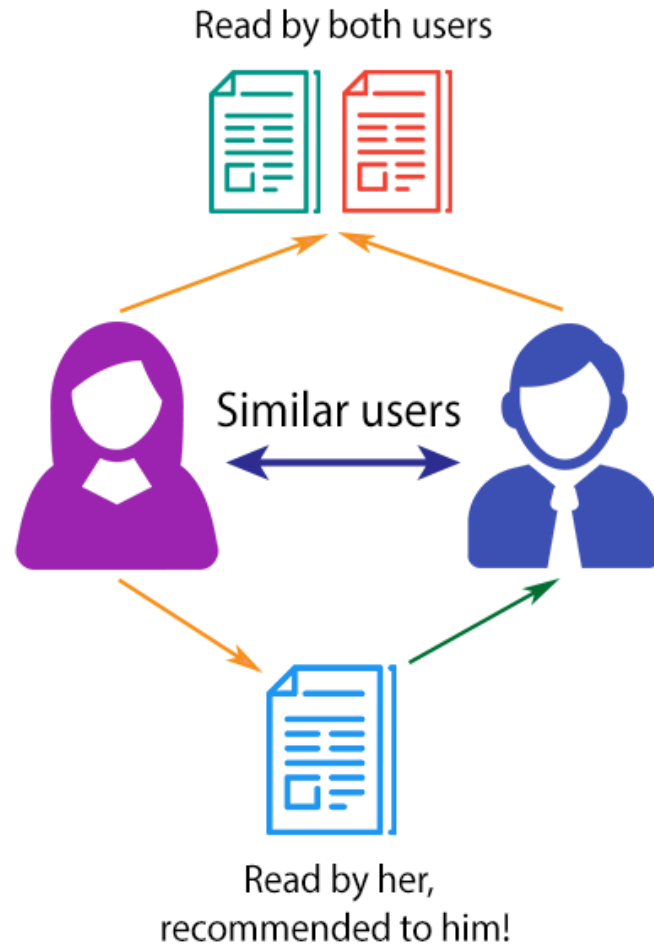
Фаза 2 • Неделя 4 • Понедельник

Рекомендательные системы • RecSys

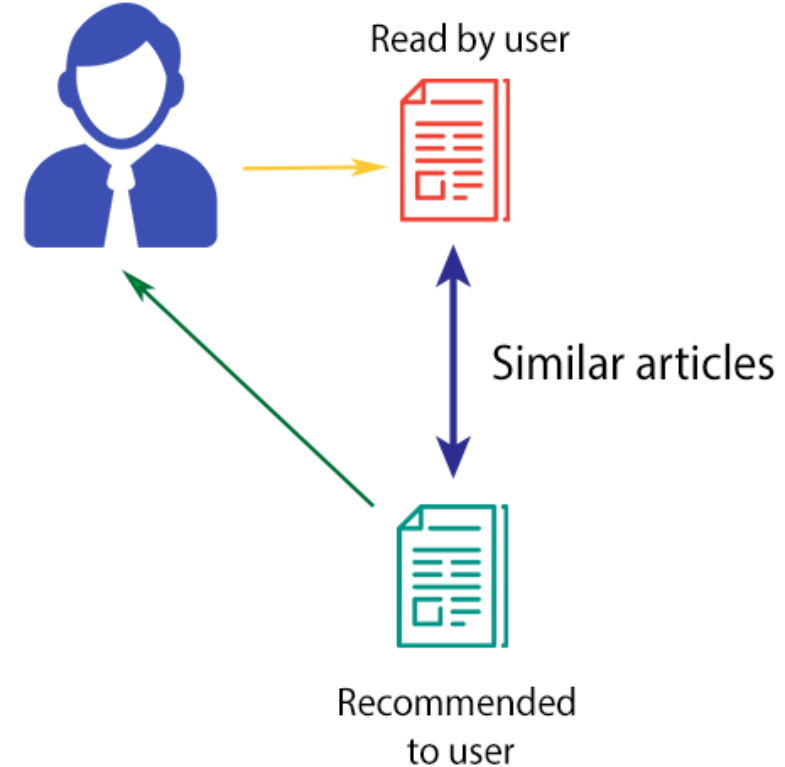
- сферы применения
- базовые методы рекомендаций
- построим рекомендательную систему на основе базы MovieLens

- Онлайн торговля
- Видеосервисы
- Музыка
- Литература
- ...

## COLLABORATIVE FILTERING



## CONTENT-BASED FILTERING



# Content-based recommendation

Цель Найти похожие объекты и рекомендовать их пользователю

## Доступная информация:

- информация о доступных продуктах
- информация о том, что и как пользователь оценивал ранее


# Как измерить близость объектов?

Измерить расстояние от оцененных пользователем объектов до  $k$  ближайших по выбранной метрике соседей и предсказать рейтинг

- косинусное сходство:  $sim(a, b) = \frac{\mathbf{a} \cdot \mathbf{b}}{|\mathbf{a}| \times |\mathbf{b}|}$

- скорректированное косинусное сходство:

$$sim(a, b) = \frac{\sum_{u \in U} (r_{u,a} - \bar{r}_u)(r_{u,b} - \bar{r}_u)}{\sqrt{\sum_{u \in U} (r_{u,a} - \bar{r}_u)^2} \sqrt{\sum_{u \in U} (r_{u,b} - \bar{r}_u)^2}}$$

- любую метрику, которая позволит определить близость объектов, подробнее можно посмотреть в документации  [scipy](#)

# User based подход

1. Найти множество пользователей, кто оценивал те же объекты, что User
2. Вычислить среднюю оценку соседей по объекту  $i$
3. Сделать это для всех объектов, которые не видел User и рекомендовать те, которые получили наибольшую среднюю оценку

user name	item 1	item 2	item 3	item 4	item 5
User	5	3	4	4	?
username 1	3	1	2	3	3
username 2	4	3	4	3	5
username 3	3	3	1	5	4
username 4	1	5	5	2	1

# User based подход

- как измерить близость?
- как много "соседей выбрать"?
- как усреднить рейтинг соседей?

user name	item 1	item 2	item 3	item 4	item 5	<i>sim</i>
User	5	3	4	4	?	
username 1	3	1	2	3	3	<i>sim</i> = .85
username 2	4	3	4	3	5	<i>sim</i> = .00
username 3	3	3	1	5	4	<i>sim</i> = .70
username 4	1	5	5	2	1	<i>sim</i> = −.79



- не все соседи могут быть одинаково ценны для предсказания оценки `User`
- можно искусственно повышать влияние близких соседей на прогнозируемый рейтинг
- на оценку близости  $sim$  можно накладывать ограничения, т.е. "фильтровать"

# Item based подход

В качестве основы для вычисления *sim* можно использовать не пользователей, а объекты ( `items` )

user name	<i>item 1</i>	item 2	item 3	<i>item 4</i>	item 5
User	5	3	4	4	?
username 1	3	1	2	3	3
username 2	4	3	4	3	5
username 3	3	3	1	5	4
username 4	1	5	5	2	1

# Проблема холодного старта

- просить пользователей оценить товар
- использовать другой подход (возможно, более грубый)
- использовать дефолтные оценки

- пользователи оценивают объекты
- пользователи, имеющие схожие вкусы в прошлом, будут иметь схожие вкусы в будущем
- используя информацию об оценках большого числа людей, можно пробовать рекомендовать объекты

# SVD в рекомендательных системах

- SVD:  $M_k = U_k \times \Sigma_k \times V_k^T$

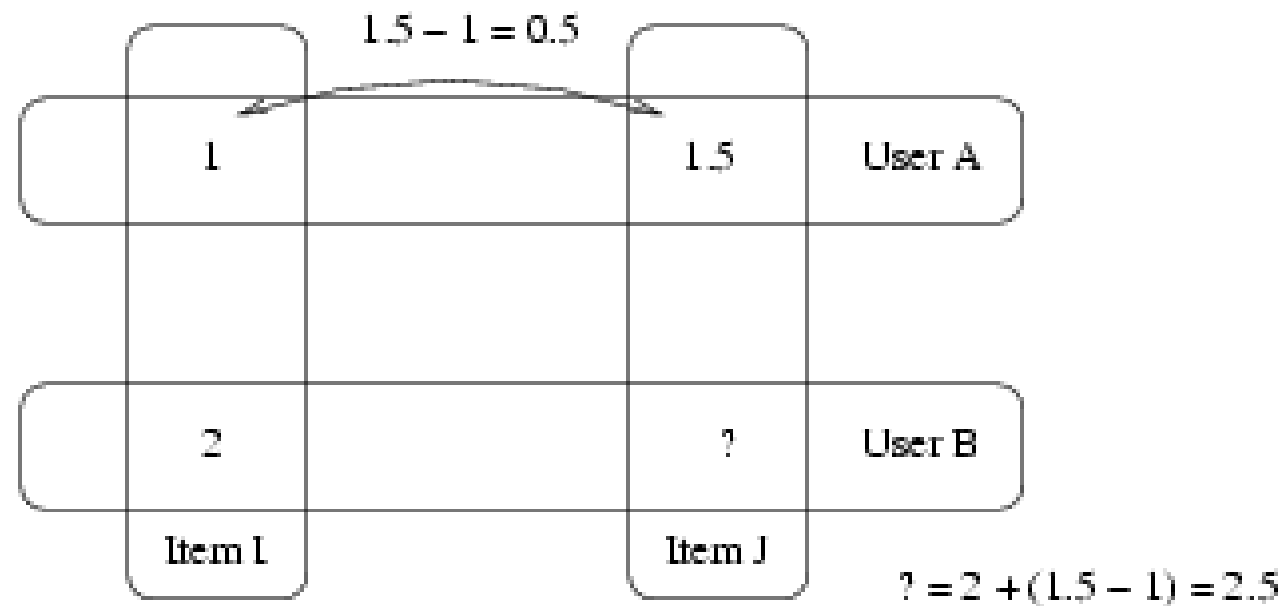
$U_k$	Dim1	Dim2
Alice	0.47	-0.30
Bob	-0.44	0.23
Mary	0.70	-0.06
Sue	0.31	0.93

$V_k^T$	Terminator	Die Hard	Twins	Eat Pray Love	Pretty Woman
Dim1	-0.44	-0.57	0.06	0.38	0.57
Dim2	0.58	-0.66	0.26	0.18	-0.36

- Prediction:  $\hat{r}_{ui} = \bar{r}_u + U_k(Alice) \times \Sigma_k \times V_k^T(EPL)$   
 $= 3 + 0.84 = 3.84$

$\Sigma_k$	Dim1	Dim2
Dim1	5.63	0
Dim2	0	3.23

# Slope one



## Slope one • Пример

	Item A	Item B	Item C
John	5	3	2
Mark	3	4	-
Lucy	???	2	5

- Найдем всех пользователей, которые оценили пару товаров
- Вычислим усредненную оценку «разницы» между двумя товарами
- Искать будем оценку пользователя **Lucy** для **Item A**

$$\text{diff}(\text{ItemA}, \text{ItemB}) = \frac{(r_{John,A} - r_{John,B}) + (r_{Mark,A} - r_{Mark,B})}{N\_pairs_{AB}} = \frac{2 - 1}{2} = 0.5$$

$$\text{diff}(\text{ItemA}, \text{ItemC}) = \frac{r_{John,A} - r_{John,C}}{N\_pairs_{AC}} = \frac{5 - 2}{1} = 3$$

## Slope one • Пример

	Item A	Item B	Item C
John	5	3	2
Mark	3	4	-
Lucy	???	2	5

$$\text{diff}(\text{ItemA}, \text{ItemB}) = 0.5$$

$$\text{diff}(\text{ItemA}, \text{ItemC}) = 3$$

$$N_{\text{pairs}}_{AB} = 2$$

$$N_{\text{pairs}}_{AC} = 1$$

Прогнозируем оценку Lucy для Item A:

- на основе ItemB:  $r_{Lucy,B} + \text{diff}(\text{ItemA}, \text{ItemB}) = 2 + 0.5 = 2.5$
- на основе ItemC:  $r_{Lucy,C} + \text{diff}(\text{ItemA}, \text{ItemC}) = 5 + 3 = 8$
- взвешенным средним:  $r_{Lucy,ItemA} = \frac{N_{\text{pairs}}_{AB} \times 2.5 + N_{\text{pairs}}_{AC} \times 8}{2+1} = 4.33$








## Slope one • Задача

	Item A	Item B	Item C
John	5	3	2
Lucy	4	2	5
Mark	3	4	?

$$r_{Mark, ItemC} = ?$$

## Еще подходы

Вероятностные подходы (включая байесовский)	 post
Основанные на кластеризации	 post
Вероятностный латентно-семантический анализ	 pdf
Naive slope one	 pdf
RF-rec predictors	 pdf

- в задачах построения рекомендаций можно и нужно проверять множество гипотез
- самый простой подход: измерять близость в векторном пространстве
- существует множество подходов в "линейных" алгоритмах

 [LightFM](#), [RecTools](#), [scikit-surprise](#)