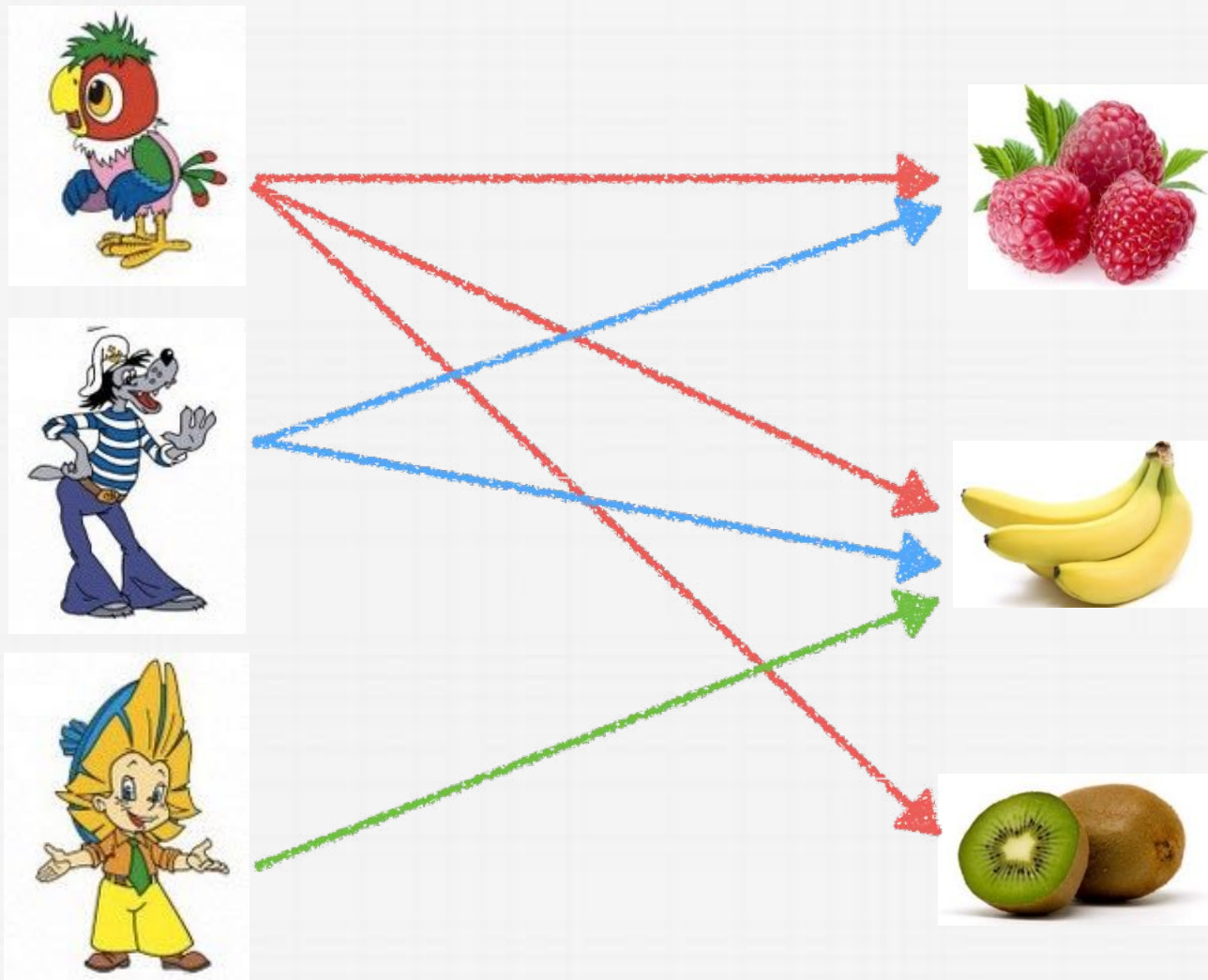




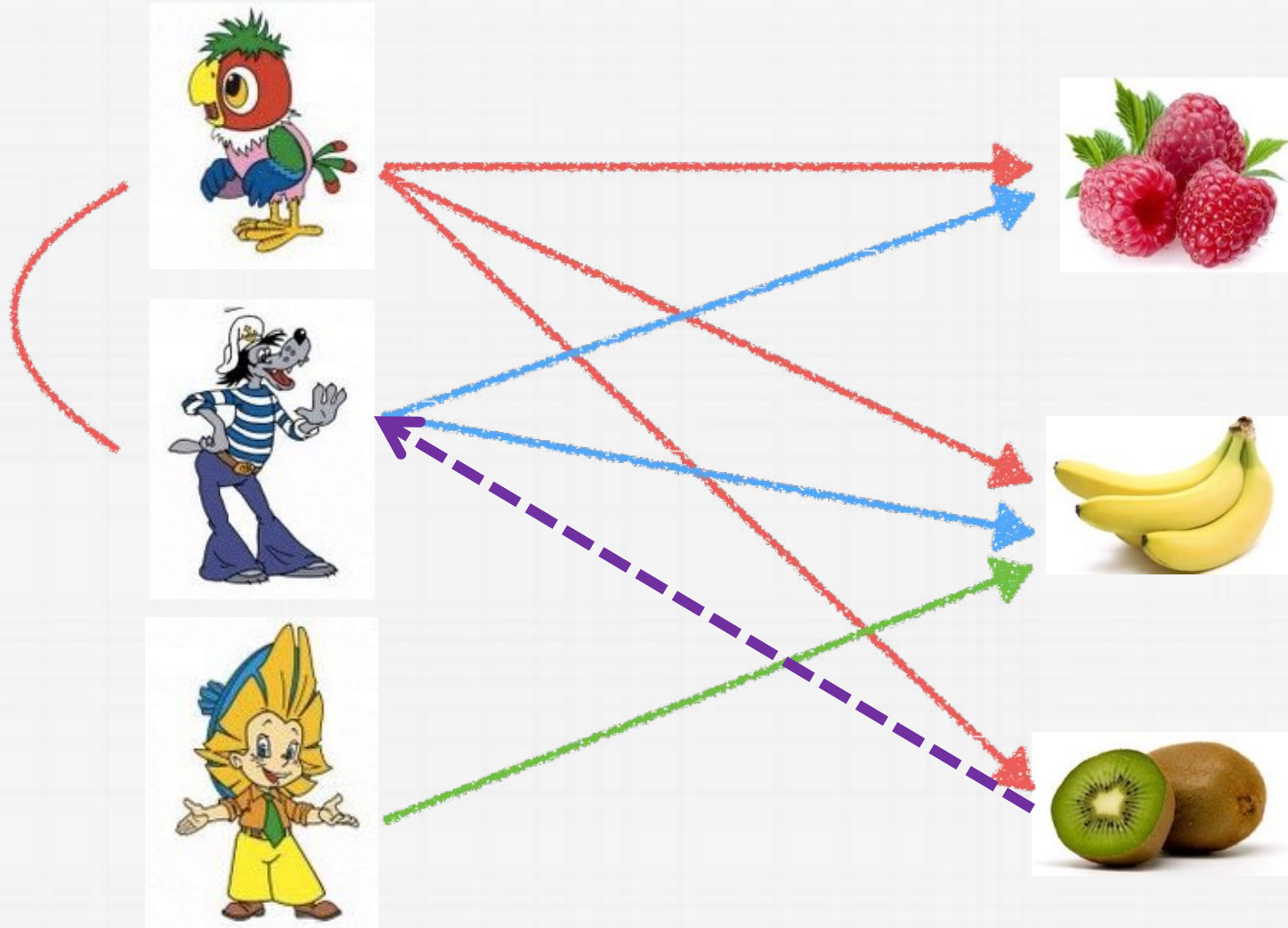
# Коллаборативная фильтрация

Андрей Зимовнов (Яндекс, ВШЭ)

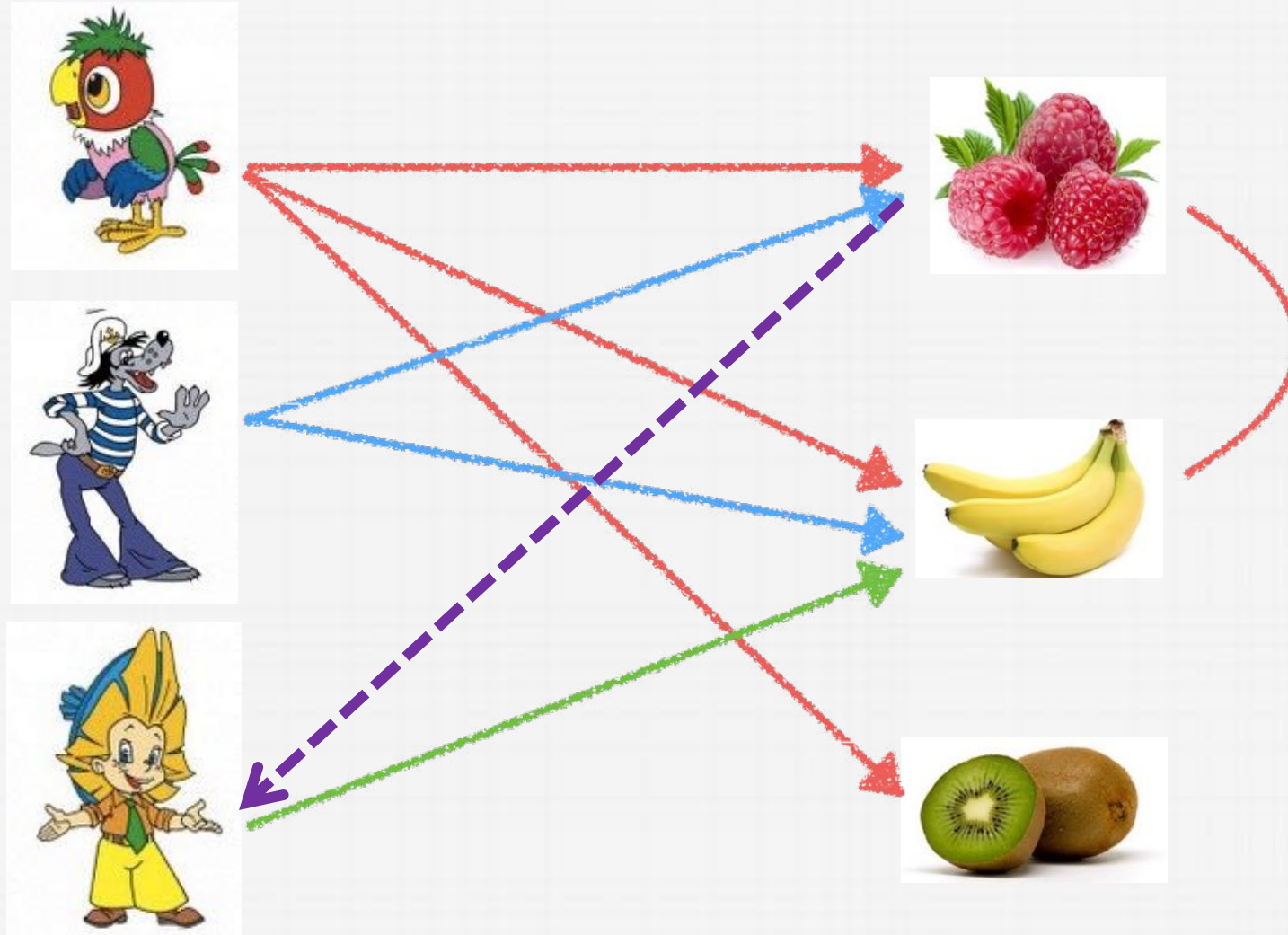
# Как использовать эти данные?



# User-based Collaborative Filtering



# Item-based CF



# Матрица оценок

Пользователи

Понравится?

	4	5	6	7	8	9
1						
2	2		2	4	5	
3	5		4			1
4			5		2	
5		1		5		4
6			4			2
7	4	5		1		

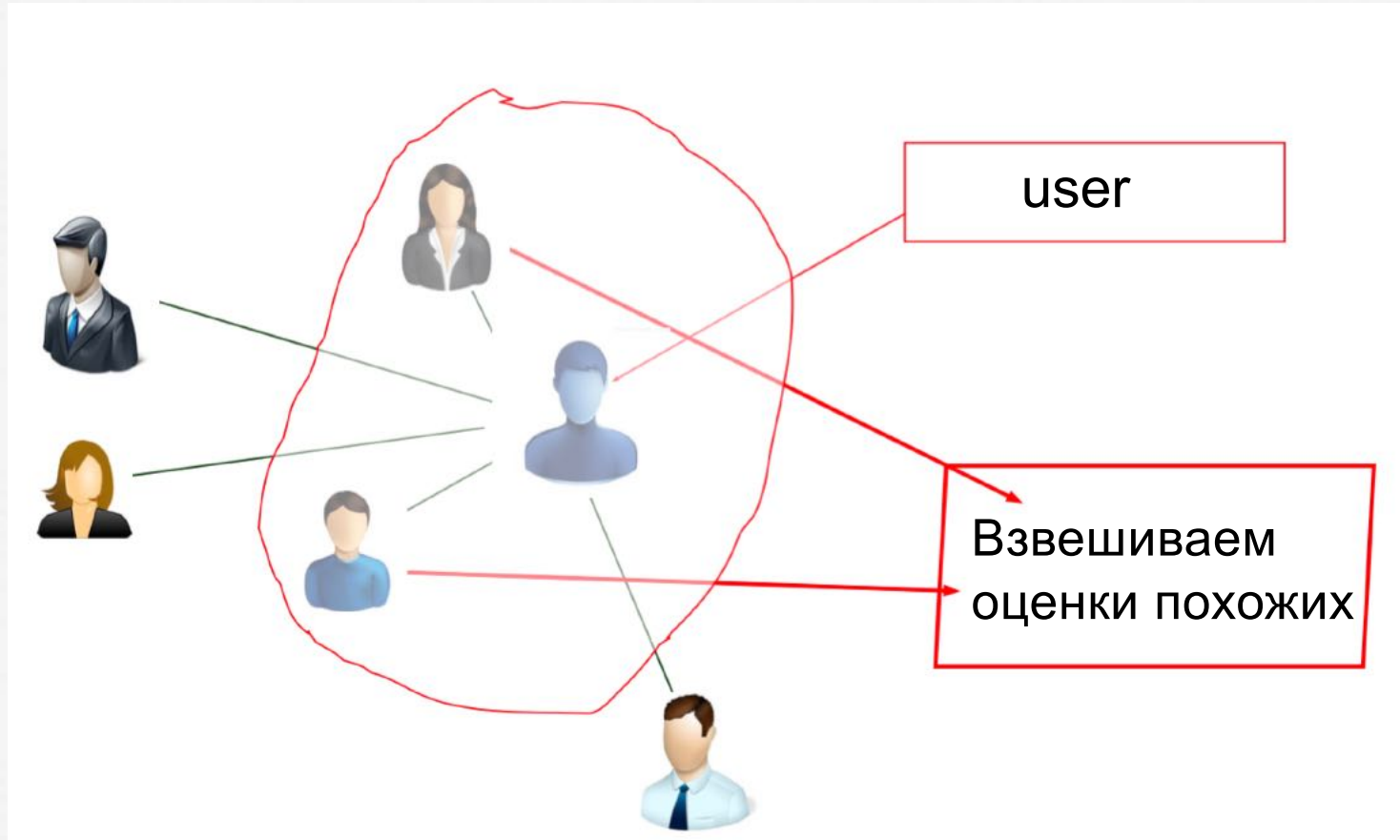
Товары

Оценка



# User-based CF

**Идея:** Найдем похожих на **user** пользователей и порекомендуем ему понравившиеся им товары.



# Что такое похожесть юзеров?

	4	5	6	7	8	9
						
	2		2	4	5	
	5		4			1
			5		2	
		1		5		4
			4			2
	4	5		1		

Идеи?

# Корреляция оценок!

средний рейтинг  
юзера (по всем  
оценкам)

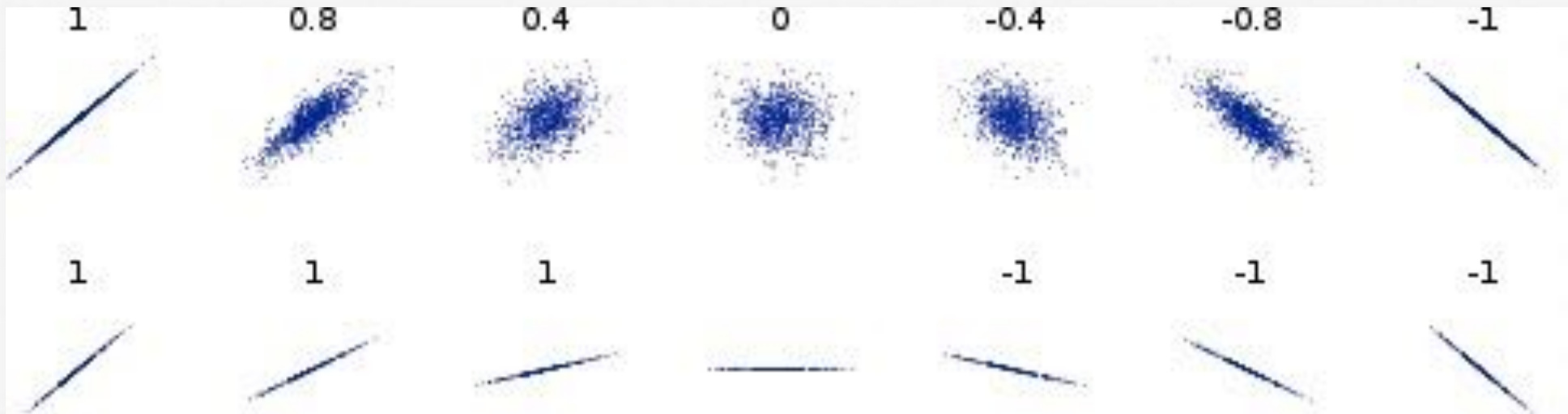
корреляция  
Пирсона

$$s(a, u) = \frac{\sum_{i \in I_a \cap I_u} (r_{ai} - \bar{r}_a)(r_{ui} - \bar{r}_u)}{\sqrt{\sum_{i \in I_a \cap I_u} (r_{ai} - \bar{r}_a)^2} \sqrt{\sum_{i \in I_a \cap I_u} (r_{ui} - \bar{r}_u)^2}}$$

общие  
рейтинги



# Корреляция Пирсона



Изменяется от -1 до 1

# Пример



user →

							sim(u,v)
	2			4	5		NA
	5		4			1	
			5		2		
		1		5		4	
			4			2	
	4	5		1			NA

Почему?

# Пример













user →

							sim(u,v)
	2			4	5		NA
	5		4			1	
			5		2		
		1		5		4	
			4			2	
	4	5		1			NA

Нет общих оценок!

# Пример













user →

							sim(u,v)
	2			4	5		NA
	5		4			1	0.87
			5		2		
		1		5		4	
			4			2	
	4	5		1			NA

Не 1, потому что  
максимальная  
оценка у юзера 5

# Пример













user →

							sim(u,v)
	2			4	5		NA
	5		4			1	0.87
			5		2		1
		1		5		4	
			4			2	
	4	5		1			NA

Если бы вычитали среднее по общим оценкам, получили бы деление на ноль!

# Пример

user →

							sim(u,v)
	2			4	5		NA
	5		4			1	0.87
			5		2		1
		1		5		4	-1
			4			2	
	4	5		1			NA

И не нашли бы это...



# Юзер с одинаковыми оценками

Если все оценки юзера одинаковые, то будет деление на ноль!

$$s(a, u) = \frac{\sum_{i \in I_a \cap I_u} (r_{ai} - \bar{r}_a)(r_{ui} - \bar{r}_u)}{\sqrt{\sum_{i \in I_a \cap I_u} (r_{ai} - \bar{r}_a)^2} \sqrt{\sum_{i \in I_a \cap I_u} (r_{ui} - \bar{r}_u)^2}}$$

↑  
0

Нужно пропускать таких пользователей!

# Мало оценок в пересечении

В случае одной общей оценки:

$$s(a, u) = \frac{\sum_{i \in I_a \cap I_u} (r_{ai} - \bar{r}_a)(r_{ui} - \bar{r}_u)}{\sqrt{\sum_{i \in I_a \cap I_u} (r_{ai} - \bar{r}_a)^2} \sqrt{\sum_{i \in I_a \cap I_u} (r_{ui} - \bar{r}_u)^2}} = \frac{(r_{ai} - \bar{r}_a)(r_{ui} - \bar{r}_u)}{\sqrt{(r_{ai} - \bar{r}_a)^2} \sqrt{(r_{ui} - \bar{r}_u)^2}} = \pm 1$$



Произведение знаков

**Проблема: большие неуверенные значения!**  
**Что делать?**

# Мало оценок в пересечении

Решение: поправочный коэффициент!

$$s(a, u) = \min\left(\frac{|I_a \cap I_u|}{50}, 1\right) \frac{\sum_{i \in I_a \cap I_u} (r_{ai} - \bar{r}_a)(r_{ui} - \bar{r}_u)}{\sqrt{\sum_{i \in I_a \cap I_u} (r_{ai} - \bar{r}_a)^2} \sqrt{\sum_{i \in I_a \cap I_u} (r_{ui} - \bar{r}_u)^2}}$$

50 – порог на количество общих рейтингов

# Что дальше?

user →

							sim(u,v)
	2			4	5		NA
	5		4			1	0.87
			5		2		1
		1		5		4	-1
			4			2	
	4	5		1			NA

Идеи?

# Среднее по соседям с весами

рейтинг  
пользователя  $u$   
для товара  $i$

мера  
близости

$$\hat{r}_{ai} = \frac{\sum_{u \in N(a)} s(a, u) r_{ui}}{\sum_{u \in N(a)} |s(a, u)|}$$

множество  
соседей

**Проблема: юзеры ставят оценки в разной шкале!**  
**Кто-то от 1 до 3, кто-то от 3 до 5! Что делать?**

# Учтем средний рейтинг!

рейтинг  
пользователя  $a$   
для товара  $i$

мера  
близости

средний  
рейтинг

$$\hat{r}_{ai} = \bar{r}_a + \frac{\sum_{u \in N(a)} s(a, u)(r_{ui} - \bar{r}_u)}{\sum_{u \in N(a)} |s(a, u)|}$$

МНОЖЕСТВО  
соседей

Проблема: Кто-то от 1 до 5, кто-то от 2 до 4! Что делать?



# И поделим на отклонение!

среднеквадратичное отклонение

$$p_{ai} = \bar{r}_a + \sigma_a \frac{\sum_{u \in N(a)} s(a, u) (r_{ui} - \bar{r}_u) / \sigma_u}{\sum_{u \in N(a)} |s(a, u)|}$$

$$\sigma_a = \sqrt{\frac{1}{m} \sum_{i=1}^m (r_{ai} - \bar{r}_a)^2}$$

Может быть отрицательным!

$$\hat{r}_{ai} = \frac{\sum_{u \in N(a)} s(a, u) r_{ui}}{\sum |s(a, u)|}$$













- Можно выкинуть отрицательные  $s(a, u)$ , теряем информацию
- Или использовать формулу с поправками (прошлый слайд)
- Или заменять отрицательные прогнозы на ближайшие неотрицательные



# Сколько соседей брать?

- Всех
- По порогу похожести
- Брать  $k$  ближайших, можно начать с  $k=30$

# Пример предсказаний

							sim(u,v)
	2			4	5		NA
	5		4			1	0.87
			5		2		1
		1		5		4	-1
	3.51*	3.81*	4	2.42*	2.48*	2	
	4	5		1			NA



# Проблемы user-based CF

- Рейтингов у юзера мало → в пересечении еще меньше → неуверенные похожести
- При появлении новой оценки похожести могут сильно измениться  
→ не получится посчитать заранее



# Прикинем на примере

- 10000 рейтингов
- 1000 пользователей
- 100 товаров
- рейтинги распределены равномерно





# Прикинем на примере

- 2 случайных пользователя в ожидании имеют 1 общий рейтинг
- 2 случайных товара в ожидании имеют 10 общих рейтингов

# Item-based CF


**Идея:** К оцененным пользователем товарам найдем наиболее похожие на них и порекомендуем.



Рекомендуем  
похожий

user оценил

# Похожесть товаров

	4	5	6	7	8	9
						
	2		2	4	5	
	5		4			1
			5		2	
		1		5		4
			4			2
	4	5		1		

Можно действовать также, как для юзеров

# Косинусная мера

$$s(i, j) = \frac{\sum_u r_{ui} r_{uj}}{\sqrt{\sum_u r_{ui}^2} \sqrt{\sum_u r_{uj}^2}}$$

Важно только  
угол между  
векторами!

Числитель считается только по общим юзерам!  
Знаменатель считается по всем юзерам!

*То есть заменили  
пропуски на нули!*

Что может пойти не так?



# Проблемка

- Рейтинги  $[1,1]$  и  $[5,5]$  считает максимально близкими!
- Нужно пропускать такие случаи



# Adjusted cosine similarity

Работает лучше (поправка на разный диапазон у юзеров):

$$s(i, j) = \frac{\sum_{u=1}^n (r_{ui} - r_u)(r_{uj} - r_u)}{\sqrt{\sum_{u=1}^n (r_{ui} - r_u)^2 \sum_{u=1}^n (r_{uj} - r_u)^2}}$$

Суммирование только по общим юзерам!








# Плюсы item-based CF

- Для популярных товаров можно получить надежную оценку похожести.
- Можно обновлять похожести товаров реже, например раз в день.

# Item-item похожести в офлайне

Будем обновлять их раз в день, считать на MapReduce при помощи инвертированного индекса (как в прошлой лекции):

						
	2			4	5	
	5		4			1
			5		2	
		1		5		4
			4			2
	4	5		1		

Вклады:

$$\begin{aligned}
 (1, 3) &\rightarrow (5 - 3.3) * (4 - 3.3) \\
 (1, 6) &\rightarrow (5 - 3.3) * (1 - 3.3) \\
 (3, 6) &\rightarrow (4 - 3.3) * (1 - 3.3)
 \end{aligned}$$

# Считаем все числители на Spark

```
def emit_pairs(x):  
    user, items = x  
    items = sorted(items)  
    if len(items) < 300:  
        for i in range(len(items) - 1):  
            for j in range(i + 1, len(items)):  
                yield (  
                    (items[i][0], items[j][0]),  
                    items[i][1] * items[j][1]  
                )  
  
dot_product = (  
    ratings  
    .map(lambda x: (x.user, (x.product, x.rating)))  
    .groupByKey()  
    .flatMap(emit_pairs)  
    .reduceByKey(lambda x, y: x + y)  
)
```

$$s(i, j) = \frac{\sum_{u=1}^n r_{ui} r_{uj}}{\sqrt{\dots}}$$



# В онлайнe

- Быстро реагируем на новые оценки пользователя
- Похожести item-item храним в Key-Value хранилище (топ ~1000 для каждого товара)



# Неявный фидбек

Для неявного фидбека, например, покупок, можно использовать меру Жаккара как похожесть!



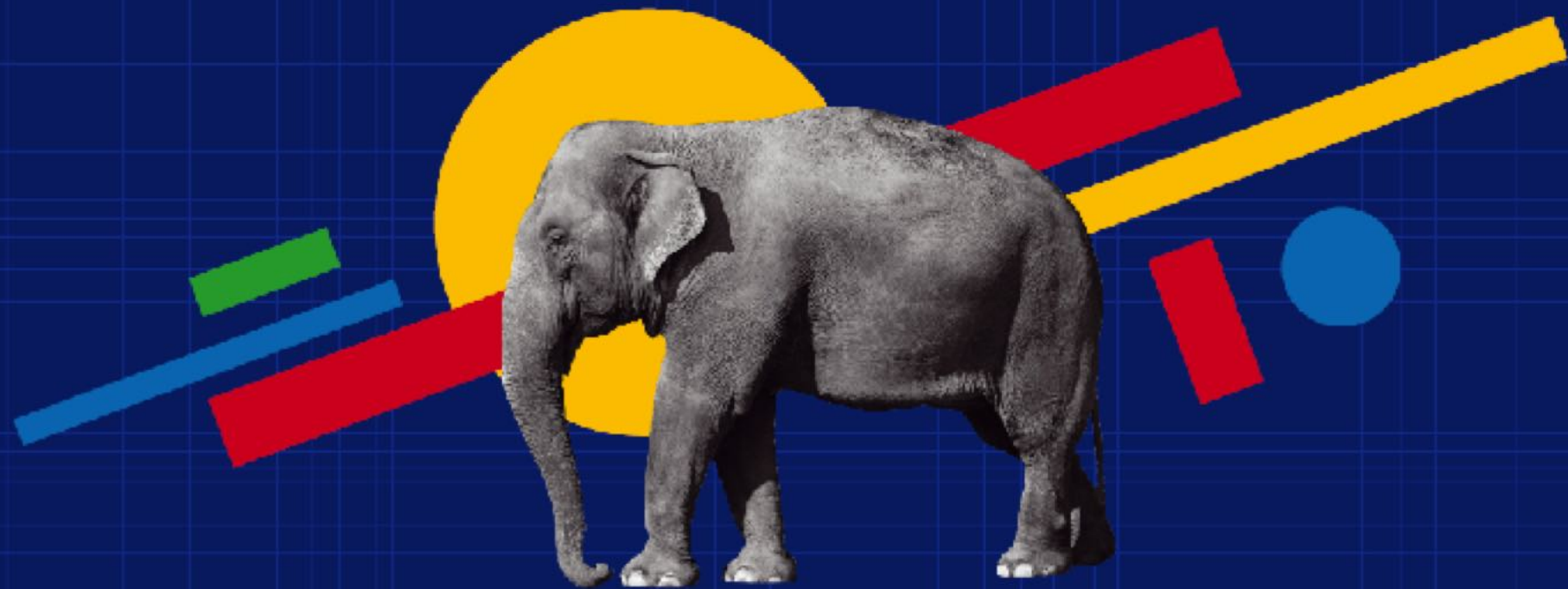
# Резюме

## Плюсы:

- Неплохие рекомендации при большом количестве явных оценок.

## Минусы:

- Плохо работает при сильной разреженности матрицы оценок
- Два пользователя должны оценивать одинаковые товары, оценка *сильно похожих* товаров не учитывается в их близости.
- Проблема холодного старта: не знаем, что делать с новым товаром или пользователем.



# BIG DATA IS LOVE

[NEWPROLAB.COM](http://NEWPROLAB.COM)