



—

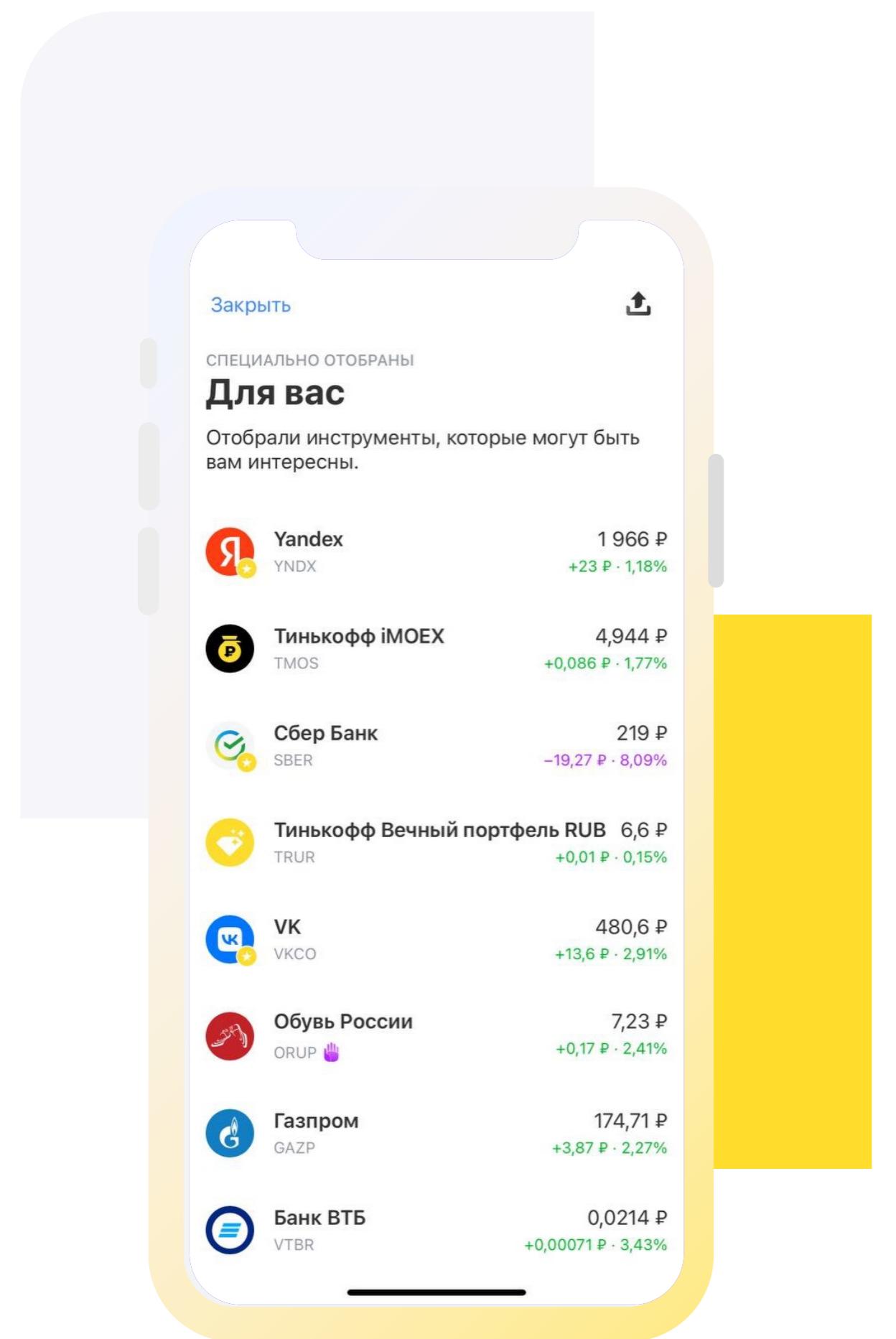
# Курс: Рекомендательные системы

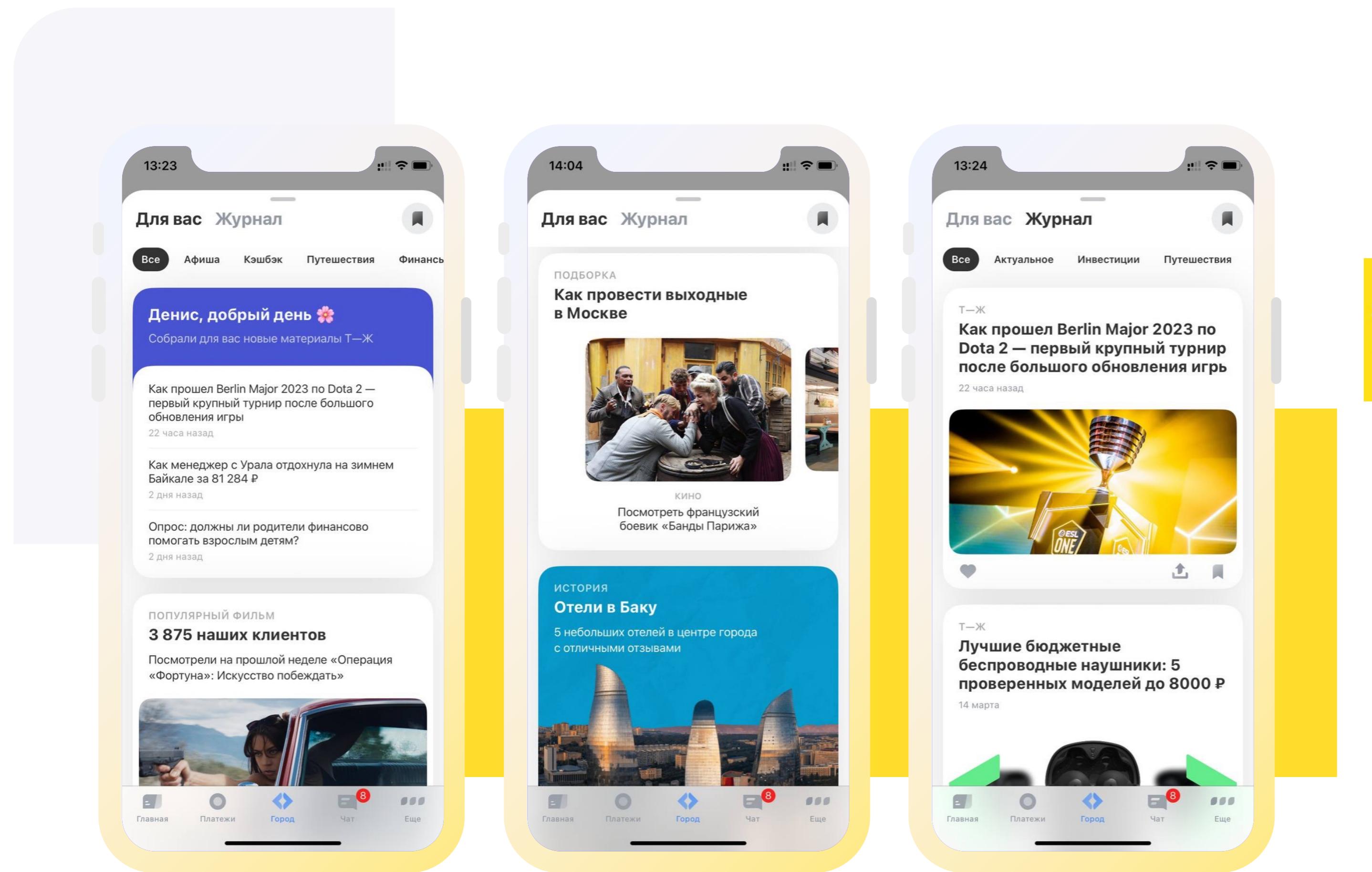
## Занятие 4:

# Коллаборитвная фильтрация

Красильников Денис  
17.02.24

# Подборка «Для Вас»





Лента

Подборки

Журнал



# Персональная лента «Для Вас»

# План занятия

- Что такое коллаборативная фильтрация?
- Семейство ALS
- Функции ранжирования
- Нейросетевые подходы

Nobody:  
YouTube Recommendations:



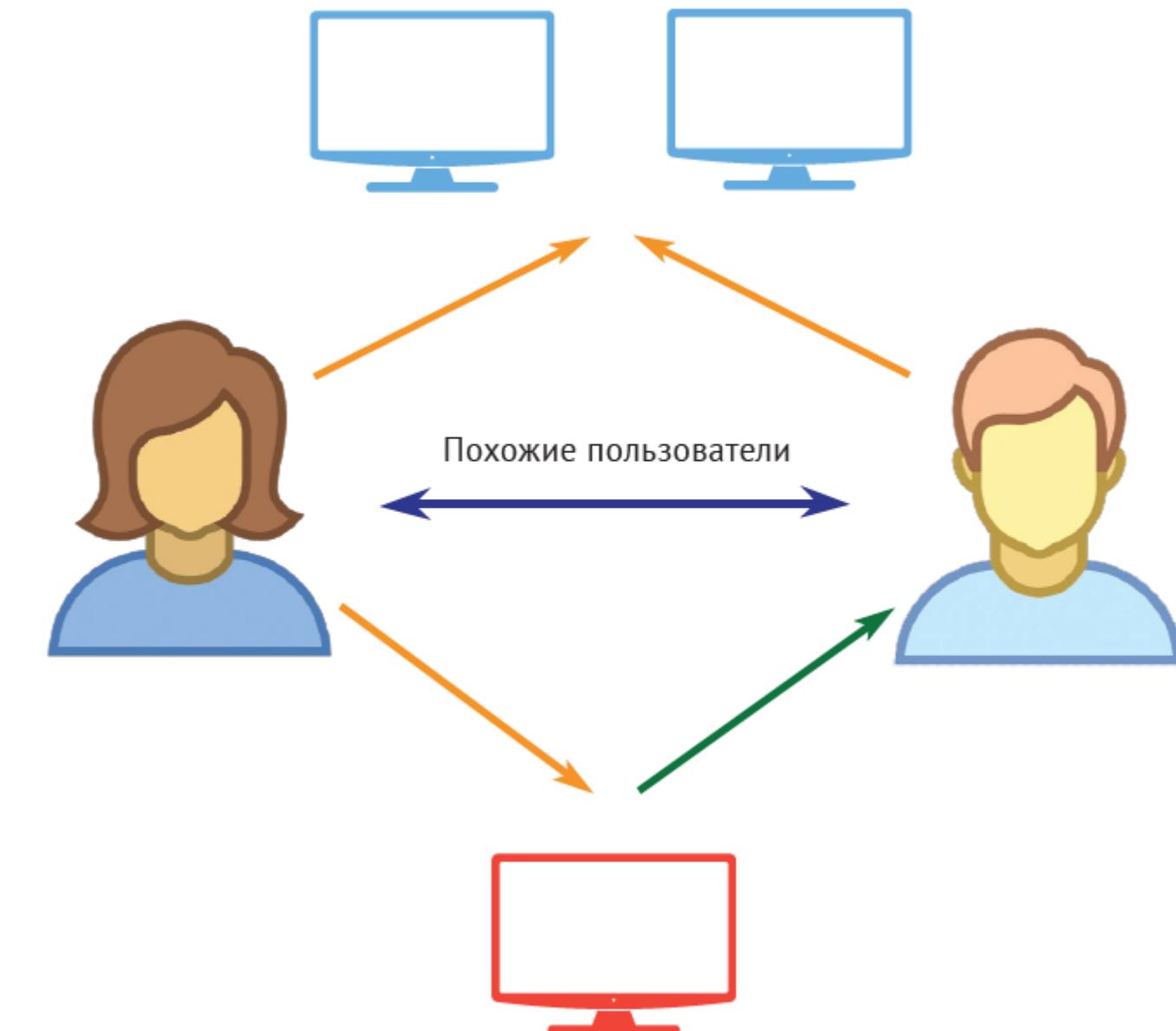
# Историческая справка



- 100,480,507 рейтингов
- 480,189 пользователей
- 17,770 фильмов
- 3 года длилось соревнование
- Призовой фонд – **1 миллион долларов**
- Победившая модель: бустинг над
  - basic model — регрессионная модель, основанная на средних оценках
  - **collaborative filtering** — колаборативная фильтрация
- RBM — ограниченные машины Больцмана
- random forests — предиктивная модель

# Коллаборативная фильтрация

Метод, использующий известные предпочтения пользователей для прогнозирования **неизвестных предпочтений** другого пользователя.  
Основное допущение состоит в следующем: те, кто одинаково оценивал какие-либо предметы в прошлом, склонны давать похожие оценки другим предметам и в будущем



Рекомендация на основе предыдущих действий пользователя

Сопоставление Смотрение Рекомендация

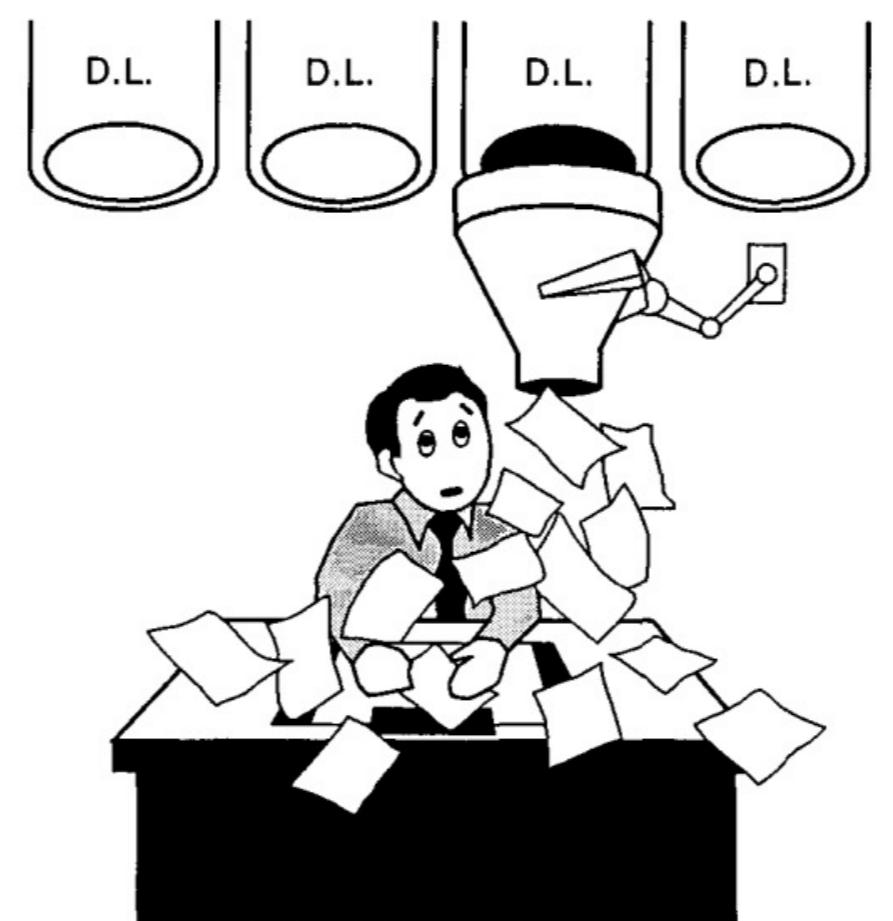
“Collaborative filtering simply means that people collaborate to help one another perform filtering by recording their reactions to documents they read”



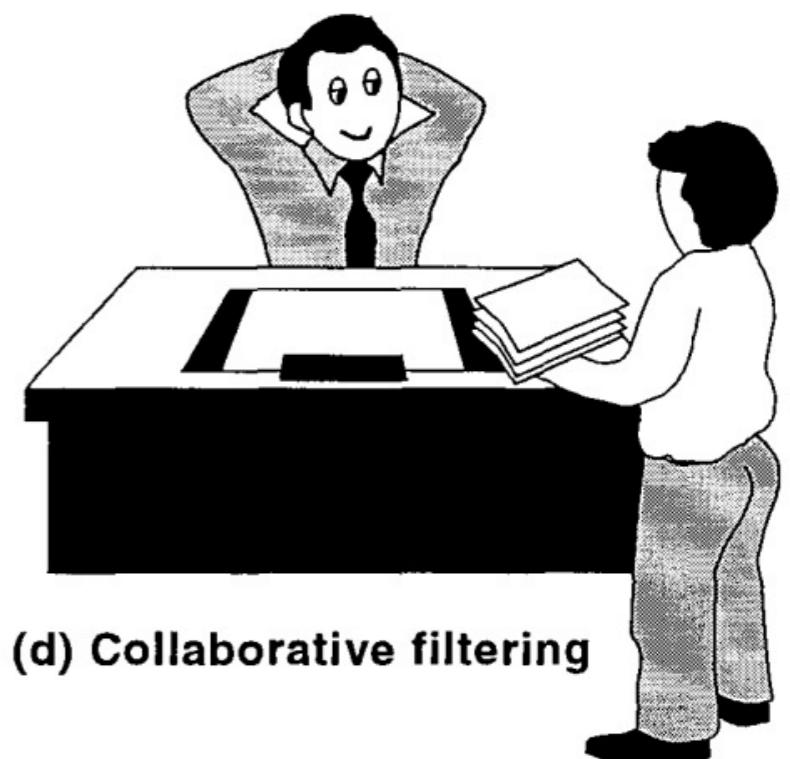
(a) Electronic mail overload



(b) Using distribution lists



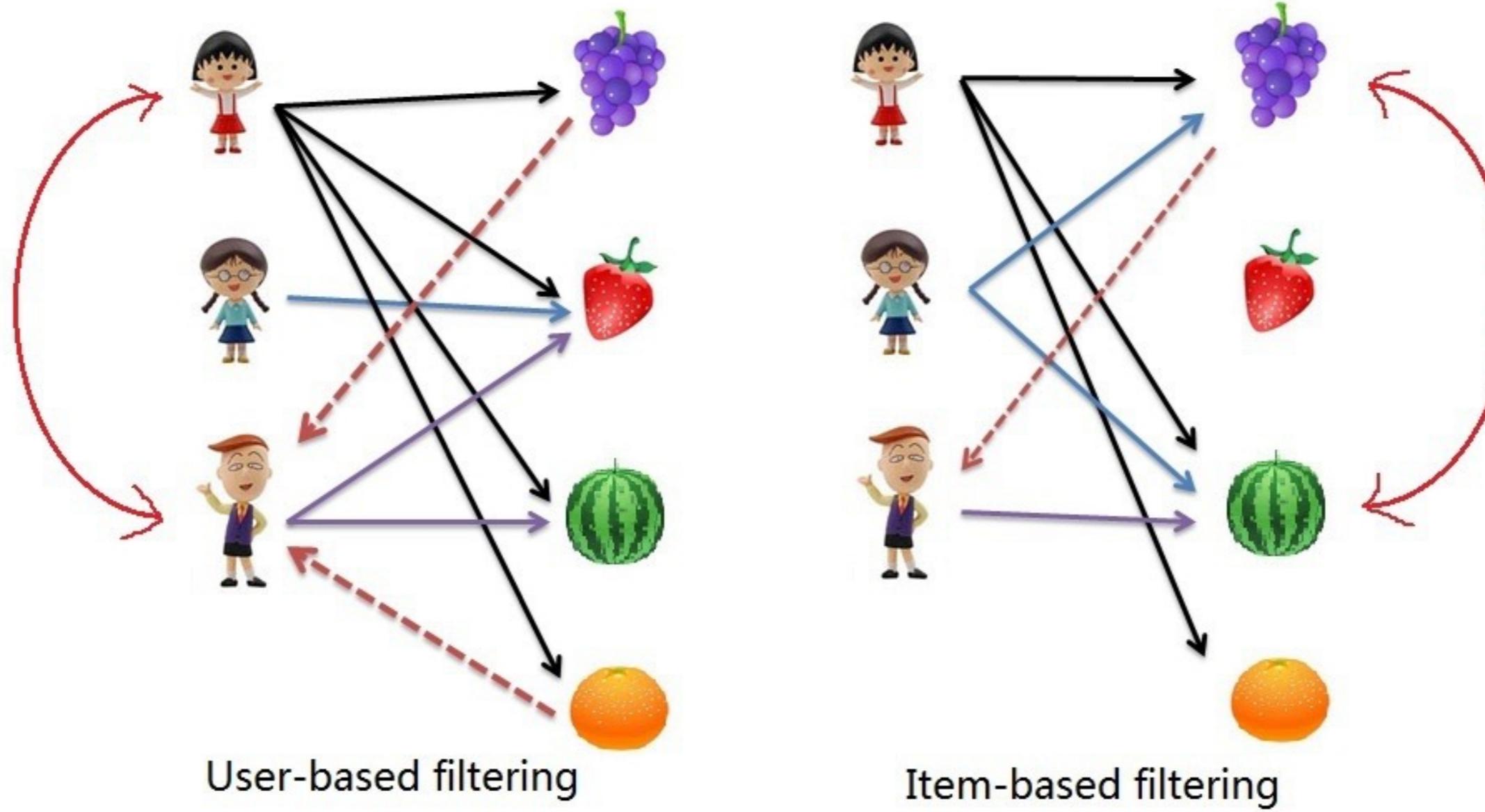
(c) Conventional filtering



(d) Collaborative filtering

CF – это идея (подход), не конкретный алгоритм!

# Коллаборативная фильтрация



# Коллаборативная фильтрация

**R =**

U/I	item 1	item2	item 3	item 3
User 1	0	3	0	1
User 2	2	0	1	0
User 3	0	5	0	1
User 4	0	3	0	0
User 5	4	0	4	5
User 6	1	2	0	4
User 7	0	3	4	0
User 8	5	0	0	5

**ИЛИ**

U/I	item 1	item2	item 3	item 3
User 1	0	1	0	1
User 2	1	0	1	0
User 3	0	1	0	1
User 4	0	1	0	0
User 5	1	0	1	1
User 6	1	1	0	1
User 7	0	1	1	0
User 8	1	0	0	1

# Коллаборативная фильтрация

## 1. Корреляционные модели (*Memory-Based Collaborative Filtering*)

- хранение всей исходной матрицы данных  $R$
- сходство клиентов - корреляция строк матрицы  $R$
- сходство объектов - корреляция столбцов матрицы  $R$

## 2. Латентные модели (*Latent Models for Collaborative Filtering*)

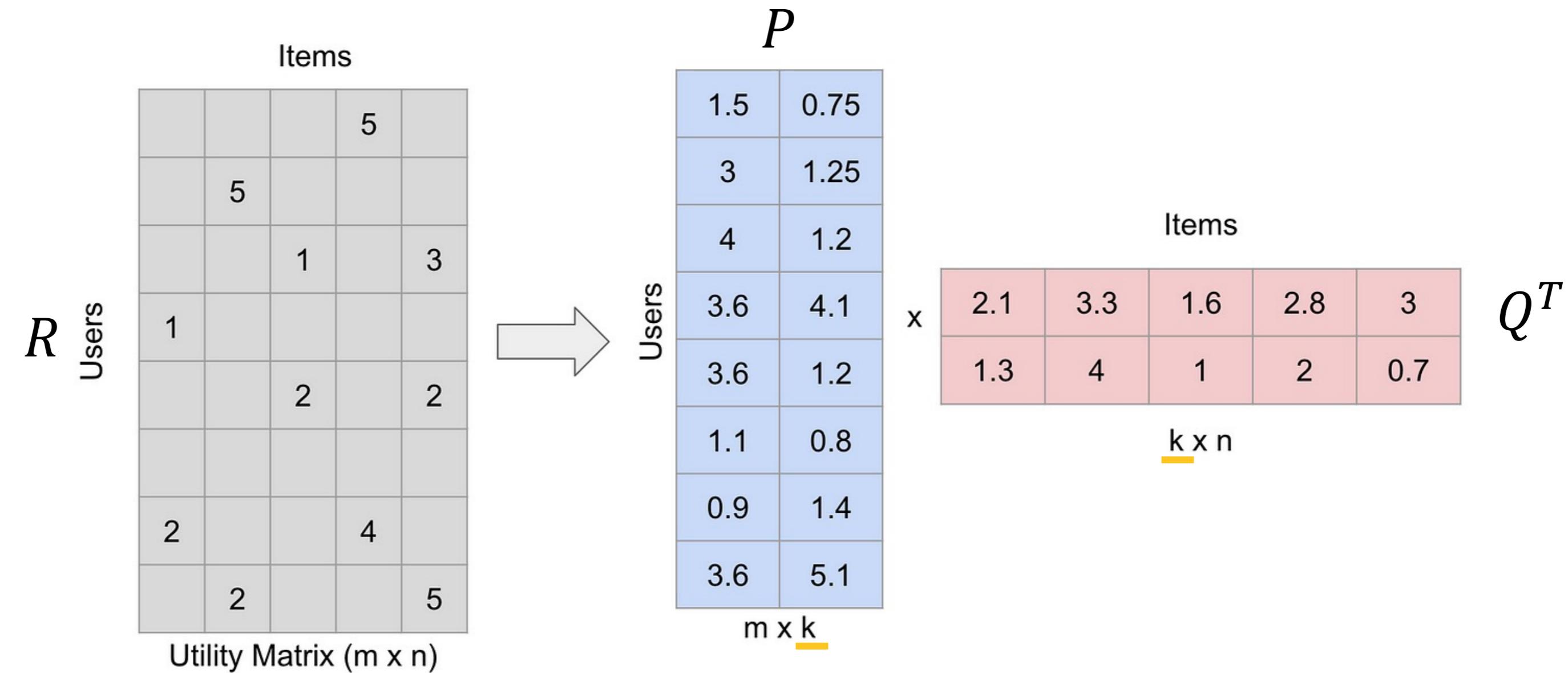
- оценивание профилей клиентов и объектов (профиль - это вектор скрытых характеристик - **эмбеддинг**)
- хранение **эмбеддингов** вместо хранения  $R$
- сходство клиентов и объектов – близость их **эмбеддингов**



# Prepare for trouble and make it double

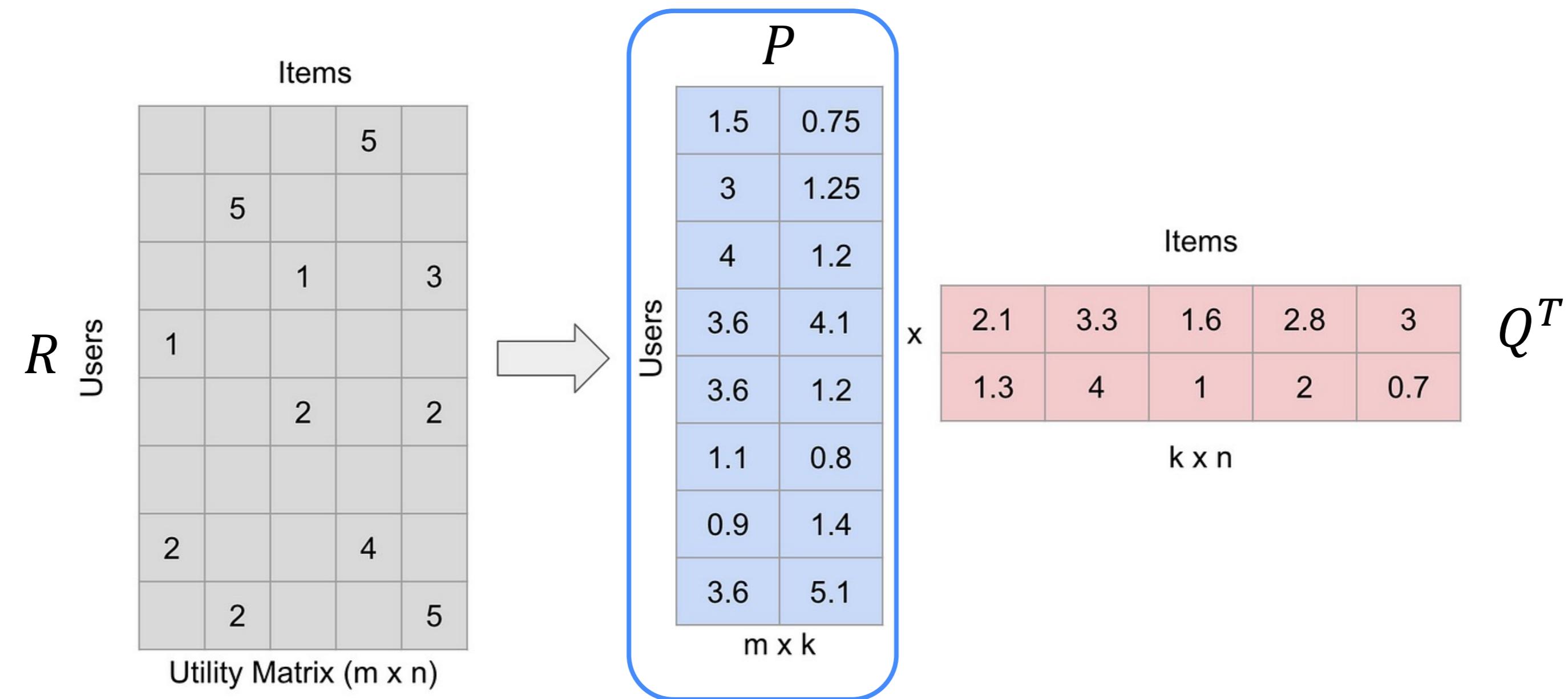
1. Холодный старт
2. Разреженность
3. Масштабируемость
4. Отсутствие разнообразия
5. «Белые вороны»

# Alternating Least Squares



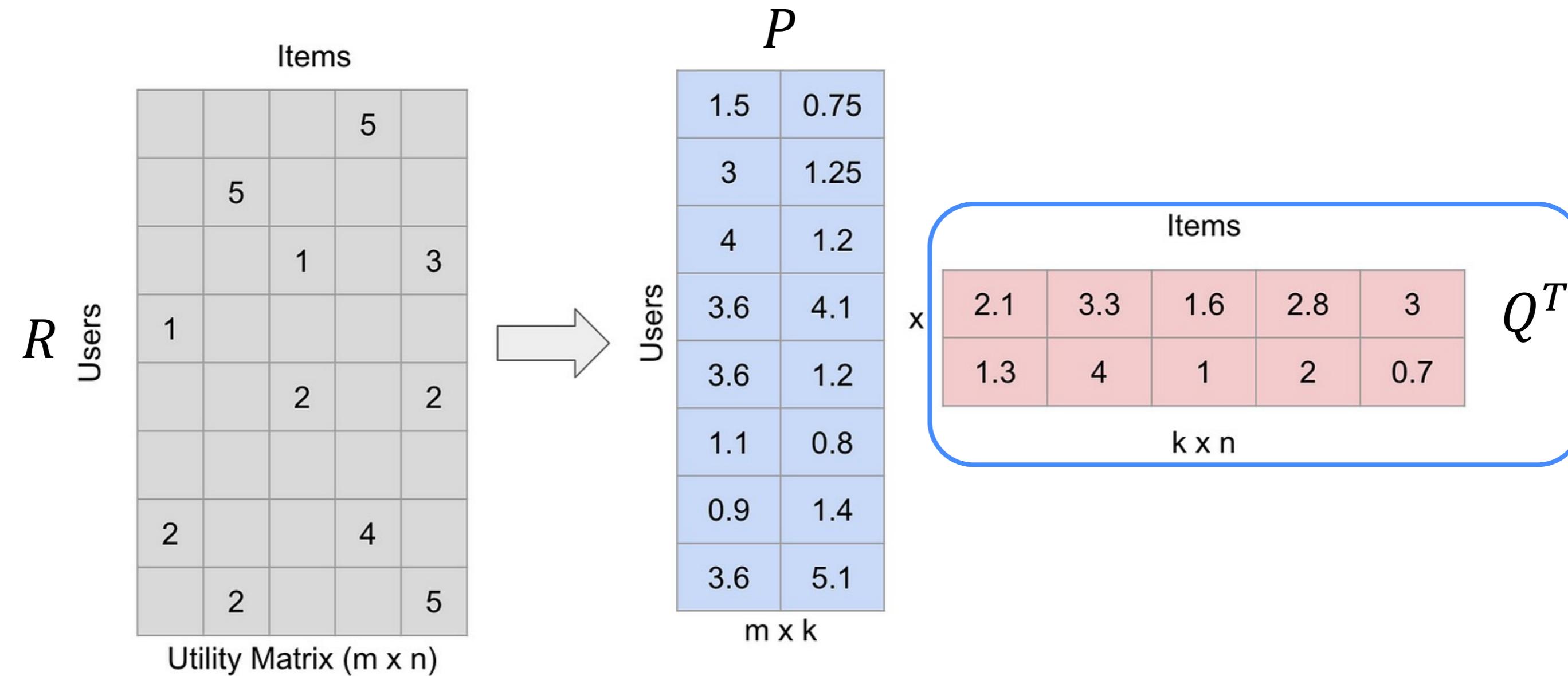
$$L = \sum_{(i,j)} \left( r_{ij} - \bar{p}_i \cdot \bar{q}_j^T \right)^2 + \alpha \left( \|\bar{p}_i\|^2 + \|\bar{q}_j\|^2 \right)$$

# Alternating Least Squares



$$L = \sum_{(i,j)} \left( r_{ij} - \bar{p}_i \cdot \bar{q}_j^T \right)^2 + \alpha \left( \|\bar{p}_i\|^2 + \|\bar{q}_j\|^2 \right)$$

# Alternating Least Squares



$$L = \sum_{(i,j)} \left( r_{ij} - \bar{p}_i \cdot \bar{q}_j^T \right)^2 + \alpha \left( \|\bar{p}_i\|^2 + \|\bar{q}_j\|^2 \right)$$

# Alternating Least Squares

$$p_{ui} = \begin{cases} 1 & r_{ui} > 0 \\ 0 & r_{ui} = 0 \end{cases}$$

$x_u$  – эмбеддинг пользователя

$y_i$  – эмбеддинг айтема

$$r_{ui} = x_u y_i^T$$

# Alternating Least Squares

$$p_{ui} = \begin{cases} 1 & r_{ui} > 0 \\ 0 & r_{ui} = 0 \end{cases}$$

$x_u$  – эмбеддинг пользователя

$y_i$  – эмбеддинг айтема

$$p_{ui} = x_u y_i^T$$

$$c_{ui} = 1 + \alpha r_{ui}$$

$$c_{ui} = 1 + \alpha \log(1 + r_{ui}/\epsilon)$$

$Y$  – матрица всех айтемов

$X$  – матрица всех пользователей

$C^u/C^i$  - диагональная матрица,  
на диагонали:  $c_{ui}$

$$x_u = (Y^T C^u Y + \lambda I)^{-1} Y^T C^u p(u)$$

$$y_i = (X^T C^i X + \lambda I)^{-1} X^T C^i p(i)$$

# Alternating Least Squares

$$p_{ui} = \begin{cases} 1 & r_{ui} > 0 \\ 0 & r_{ui} = 0 \end{cases}$$

$x_u$  – эмбеддинг пользователя

$y_i$  – эмбеддинг айтема

$$p_{ui} = x_u y_i^T$$

$$\beta = (\mathbf{X}^T \mathbf{X})^{-1} (\mathbf{X}^T \mathbf{Y})$$

$$c_{ui} = 1 + \alpha r_{ui}$$

$$c_{ui} = 1 + \alpha \log(1 + r_{ui}/\epsilon)$$

$Y$  – матрица всех айтемов

$X$  – матрица всех пользователей

$C^u/C^i$  - диагональная матрица,  
на диагонали:  $c_{ui}$

$$x_u = (Y^T C^u Y + \lambda I)^{-1} Y^T C^u p(u)$$

$$y_i = (X^T C^i X + \lambda I)^{-1} X^T C^i p(i)$$

# Alternating Least Squares

---

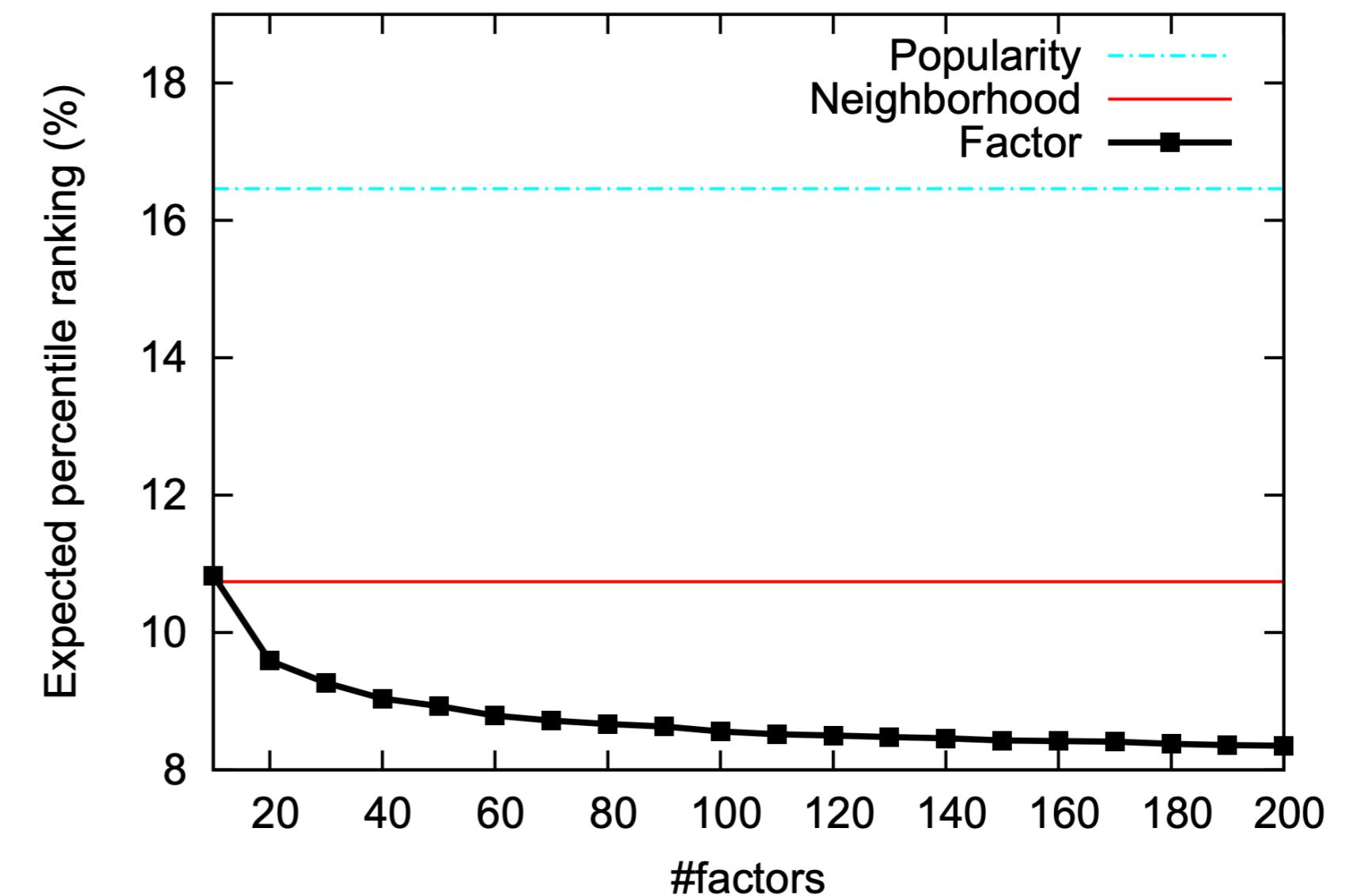
**Algorithm 1** The ALS algorithm
 

---

```

1: procedure ALS( $R, f, \lambda; X, Y$ )
2:    $X \leftarrow 0, Y \leftarrow \text{random initial guess}$ 
3:   repeat
4:     for row  $u \leftarrow 1, m$  do
5:        $x_u \leftarrow (Y^T Y + \lambda I)^{-1} Y^T r_u$  by solving the linear system
6:        $(Y^T Y + \lambda I)x_u = Y^T r_u$ 
7:     end for
8:     for column  $i \leftarrow 1, n$  do
9:        $y_i \leftarrow (X^T X + \lambda I)^{-1} X^T r_i$  by solving the linear system
10:       $(X^T X + \lambda I)y_i = X^T r_i$ 
11:    end for
12:   until reaching the maximum iterations
13: end procedure
  
```

---



Hu, Yifan, Yehuda Koren, and Chris Volinsky. "Collaborative filtering for implicit feedback datasets." *2008 Eighth IEEE international conference on data mining*. Ieee, 2008.

# Бонус: объяснения Alternating Least Squares

Рекомендуем посмотреть «Хищник», потому что смотрели «Чужой»

	N@1	N@3	N@10	R@1	R@3	R@10	M@1	M@5	M@10
MOSTPOP	0.34	0.54	0.63	0.22	0.65	0.84	0.34	0.46	0.52
RANDOM	0.37	0.53	0.62	0.23	0.63	0.82	0.37	0.45	0.51
$AR_{support}$	0.38	0.55	0.64	0.24	0.65	0.84	0.38	0.47	0.53
CHRONO	0.4	0.55	0.64	0.25	0.64	0.83	0.4	0.47	0.53
$AR_{confidence}$	0.39	0.55	0.64	0.25	0.64	0.83	0.39	0.47	0.53
SIMCATMOSTPOP	0.4	0.56	0.64	0.26	0.64	0.83	0.4	0.48	0.54
EASE	0.42	0.58	0.66	0.27	0.66	0.84	0.42	0.5	0.56
PERSONPOP	0.42	0.58	0.66	0.27	0.67	0.85	0.42	0.5	0.56
$AR_{leverage}$	0.46	0.59	0.66	0.29	0.66	0.84	0.46	0.51	0.56
SIMCATIALS	0.5	0.62	0.7	0.3	0.7	0.87	0.5	0.54	0.6
IALS	<b>0.58</b>	<b>0.71</b>	<b>0.76</b>	<b>0.37</b>	<b>0.77</b>	<b>0.91</b>	<b>0.58</b>	<b>0.64</b>	<b>0.68</b>

# Бонус: объяснения Alternating Least Squares

$$p_{ui} = y_i^T x_u$$

$$x_u = (Y^T C^u Y + \lambda I)^{-1} Y^T C^u p(u)$$

$$y_i^T (Y^T C^u Y + \lambda I)^{-1} Y^T C^u p(u)$$

$$W^u := (Y^T C^u Y + \lambda I)^{-1}$$

$$s_{ij}^u = y_i^T W^u y_j$$

# Функции ранжирования



# Функции ранжирования

## 1. Pointwise (точечные)

Например, решение через задачу регрессии или классификации.

$$\sum_{i=1}^l (a(x_i) - y(x_i))^2 \rightarrow \min$$

# Функции ранжирования

1. Pointwise (точечные)

$$\sum_{x_i < x_j} [a(x_j) - a(x_i) < 0] \rightarrow \min$$

2. Pairwise (попарные)

# Функции ранжирования

1. Pointwise (точечные)

$$\sum_{x_i < x_j} [a(x_j) - a(x_i) < 0] \rightarrow \min$$

2. Pairwise (попарные)

$$\sum_{x_i < x_j} L(a(x_j) - a(x_i)) \rightarrow \min$$

# Функции ранжирования

1. Pointwise (точечные)

**RankNet<sup>[1]</sup>:**

$$\omega := \omega + \eta \frac{1}{1+exp(\langle \omega, x_j - x_i \rangle)} (x_j - x_i)$$

2. Pairwise (попарные)

**LambdaRank<sup>[2]</sup>:**

$$\omega := \omega + \eta \frac{1}{1+exp(\langle \omega, x_j - x_i \rangle)} (x_j - x_i) |\Delta nDCG_{ij}| (x_j - x_i)$$

3. Listwise (списочные)

[1] Burges C, Shaked T, Renshaw E, Lazier A, Deeds M, Hamilton N, Hullender G. Learning to rank using gradient descent.

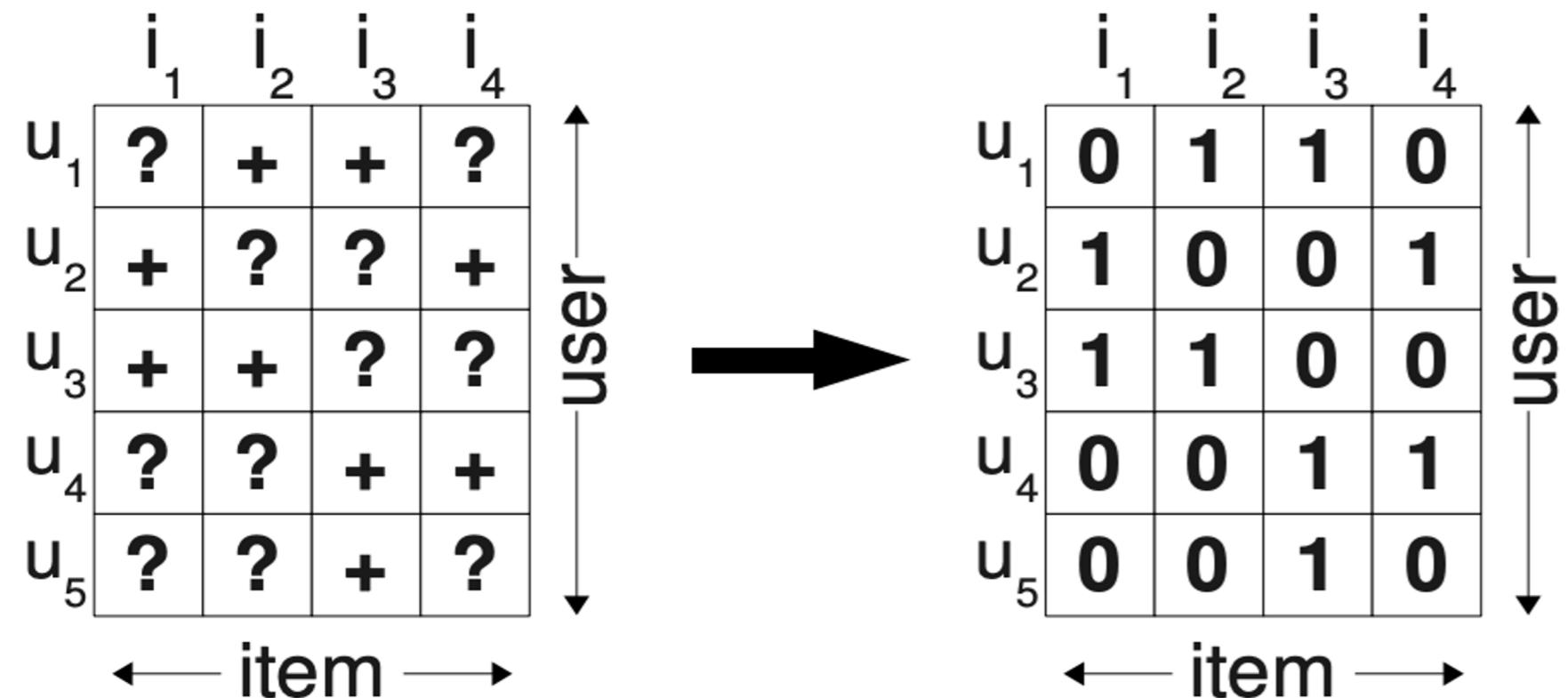
[2] C.J.C. Burges, R. Ragno and Q.V. Le. Learning to Rank with Non-Smooth Cost Functions.

# ФУНКЦИИ РАНЖИРОВАНИЯ

- **BPR** (Bayesian Personalised Ranking, pairwise)
- **WARP** (Weighted Approximate-Rank Pairwise)
- **RankNET** (pairwise/listwise)
- **LambdaRANK** (pairwise/listwise)
- **LambdaMART<sup>[1]</sup>** (pairwise/listwise)

[1] Adapting Boosting for Information Retrieval Measures. Journal of Information Retrieval, 2007.

# BPR: Bayesian Personalized Ranking



$positive >_{user} unknown$

•Рис. 1: Бинаризация implicit данных, где отсутствие интеракции кодируется как отрицательный пример.

# BPR: Bayesian Personalized Ranking

	$i_1$	$i_2$	$i_3$	$i_4$
$u_1$	?	+	+	?
$u_2$	+	?	?	+
$u_3$	+	+	?	?
$u_4$	?	?	+	+
$u_5$	?	?	+	?

*positive ><sub>user</sub> unknown*



	$i_1$	$i_2$	$i_3$	$i_4$
$u_1$	0	1	1	0
$u_2$	1	0	0	1
$u_3$	1	1	0	0
$u_4$	0	0	1	1
$u_5$	0	0	1	0

*user*

*item*

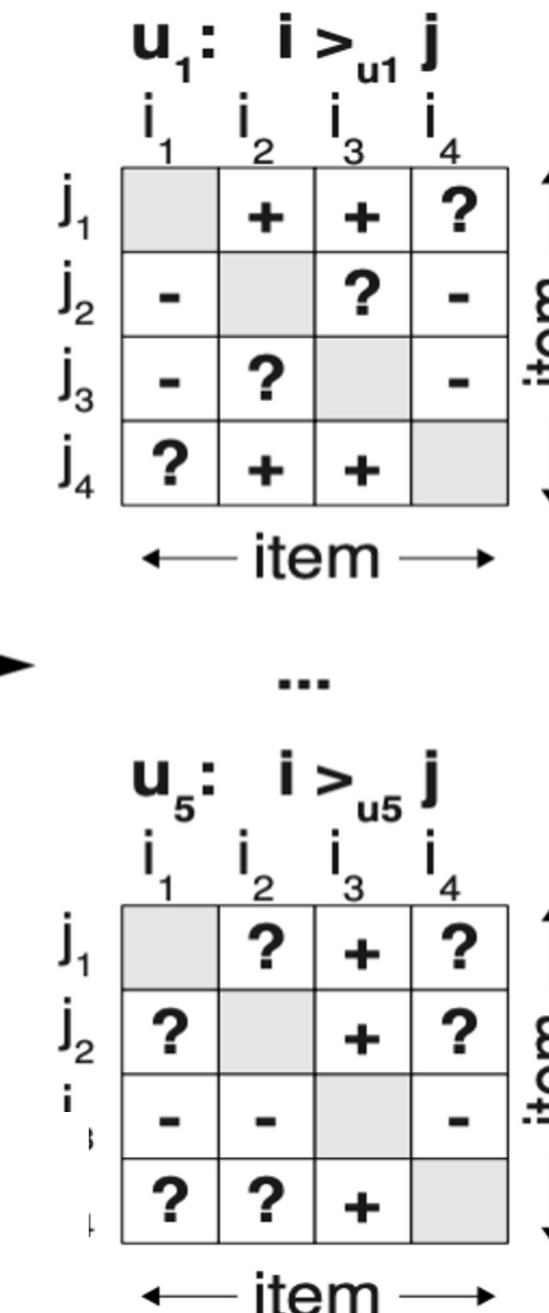
	$i_1$	$i_2$	$i_3$	$i_4$
$u_1$	?	+	+	?
$u_2$	+	?	?	+
$u_3$	+	+	?	?
$u_4$	?	?	+	+
$u_5$	?	?	+	?

*user*

*item*

$$L_{BPR}(u, i, j) = 1 - \sigma(\hat{r}_{ui} - \hat{r}_{uj})$$

$$L_{BPR}(u, i, j) = \ln \sigma(\hat{r}_{ui} - \hat{r}_{uj})$$



**Рис. 1:** Бинаризация implicit данных, где отсутствие интеракции кодируется как отрицательный пример.

**Рис. 2:** Попарные предпочтения пользователю по объектам для BPR. + означает, что пользователь предпочитает i объект j-му, знак минуса - наоборот.

# WARP:Weighted Approximate-Rank Pairwise\*

- Тоже смотрим на тройки: (**user, positive\_item, negative\_item**)
- Делаем шаг оптимизации, только если попалась дефектная пара - алгоритм дал выше скор отрицательному примеру, чем положительные
- Добавляем adaptive learning rate: штрафуем больше, если алгоритм ошибается почти сразу.

---

**Algorithm 1** K-os algorithm for picking a positive item.
 

---

We are given a probability distribution  $P$  of drawing the  $i^{th}$  position in a list of size  $K$ . This defines the choice of loss function.

Pick a user  $u$  at random from the training set.

Pick  $i = 1, \dots, K$  positive items  $d_i \in \mathcal{D}_u$ .

Compute  $f_{d_i}(u)$  for each  $i$ .

Sort the scores by descending order, let  $o(j)$  be the index into  $d$  that is in position  $j$  in the list.

Pick a position  $k \in 1, \dots, K$  using the distribution  $P$ .

Perform a learning step using the positive item  $d_{o(k)}$ .

---



---

**Algorithm 2** K-os WARP loss
 

---

Initialize model parameters (mean 0, std. deviation  $\frac{1}{\sqrt{m}}$ ).

**repeat**

Pick a positive item  $d$  using Algorithm 1.

Set  $N = 0$ .

**repeat**

Pick a random item  $\bar{d} \in \mathcal{D} \setminus \mathcal{D}_u$ .

$N = N + 1$ .

**until**  $f_{\bar{d}}(u) > f_d(u) - 1$  or  $N \geq |\mathcal{D} \setminus \mathcal{D}_u|$

**if**  $f_{\bar{d}}(y) > f_d(u) - 1$  **then**

Make a gradient step to minimize:

$$\Phi\left(\frac{|\mathcal{D} \setminus \mathcal{D}_u|}{N}\right) \max(0, 1 + f_{\bar{d}}(u) - f_d(u)).$$

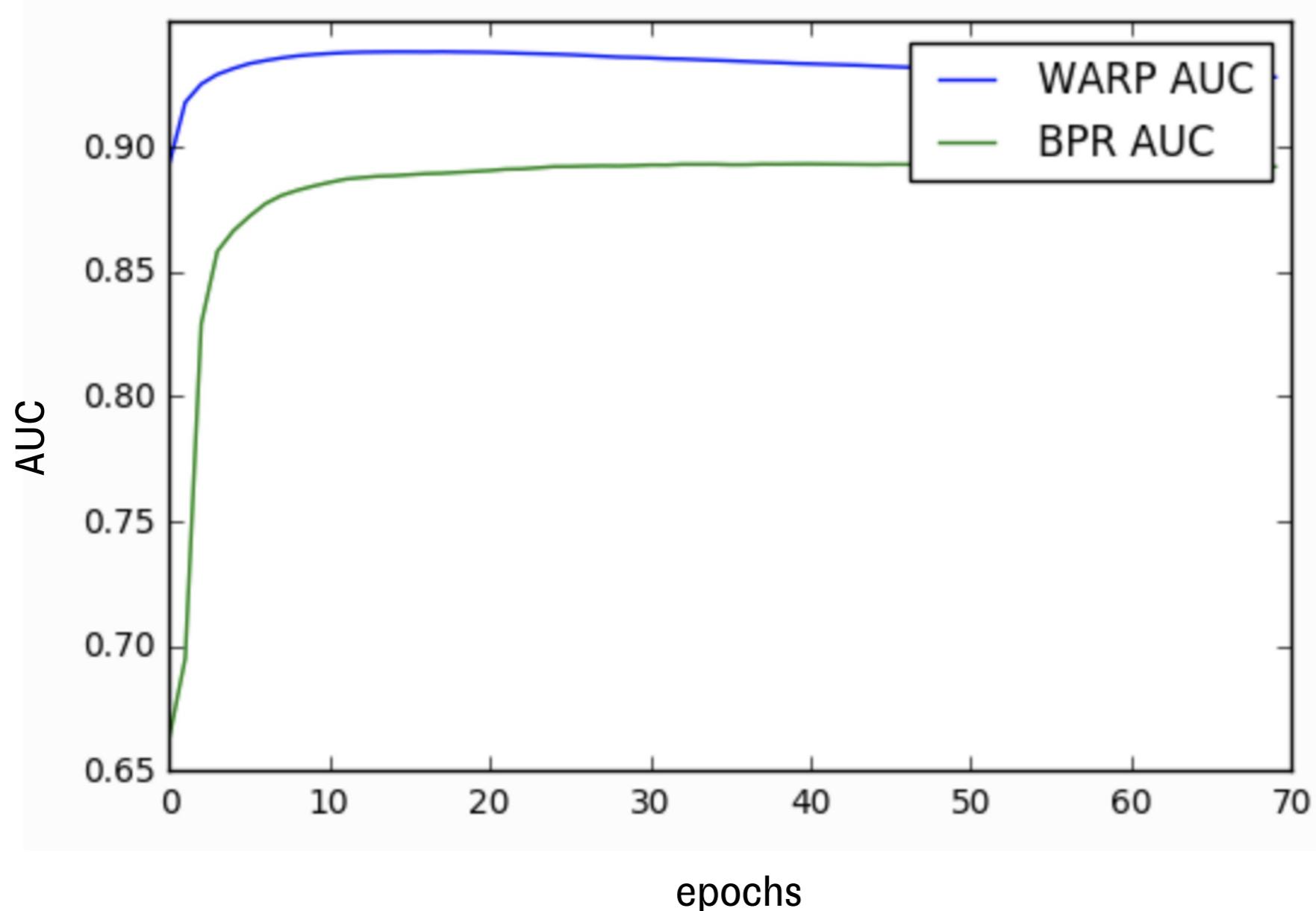
Project weights to enforce constraints, e.g. if  $\|V_i\| > C$  then set  $V_i \leftarrow (CV_i)/\|V_i\|$ .

**end if**

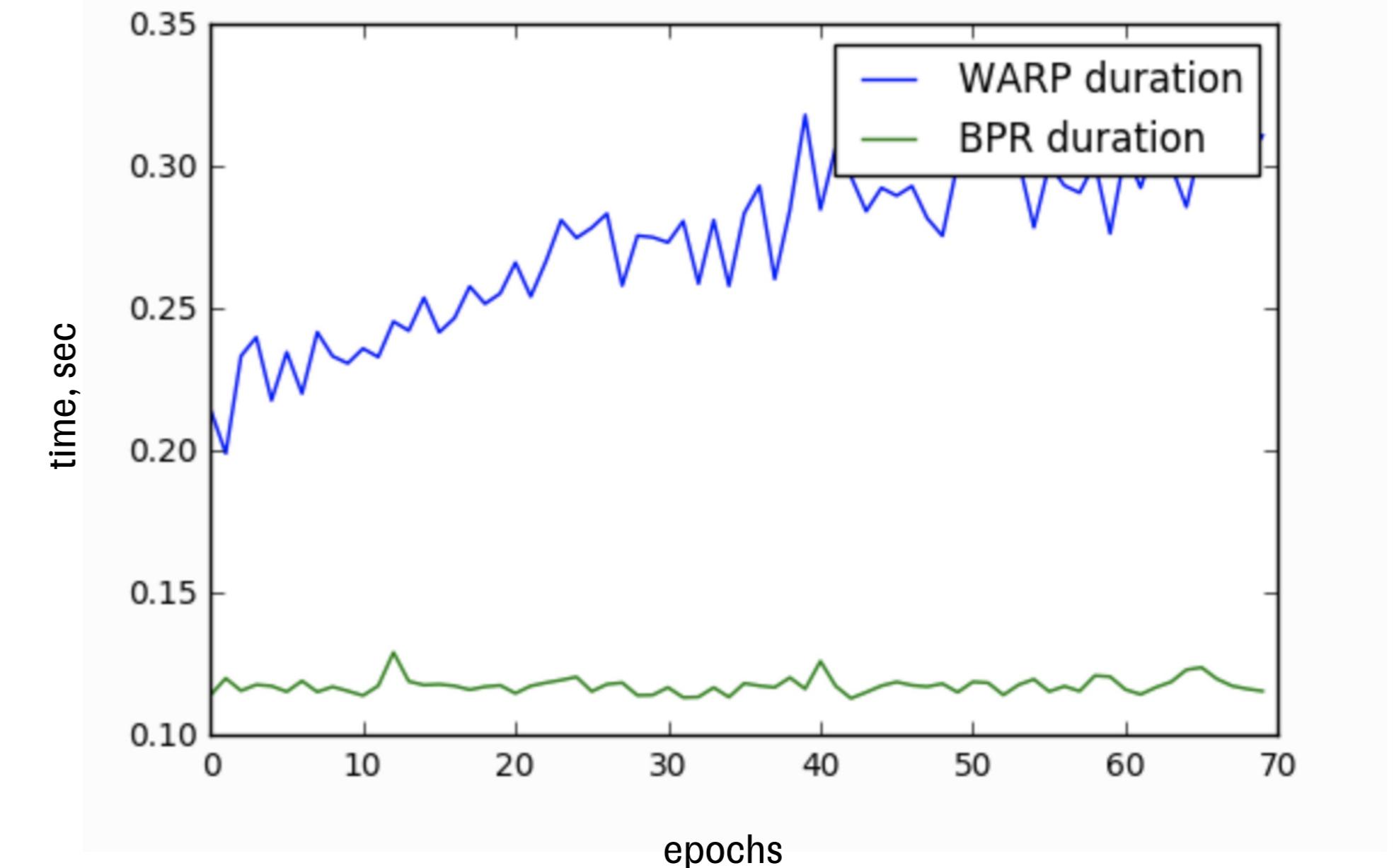
**until** validation error does not improve.
 

---

# WARP vs. BPR\*



Обычно, WARP лучше по качеству...



но более медленный =(

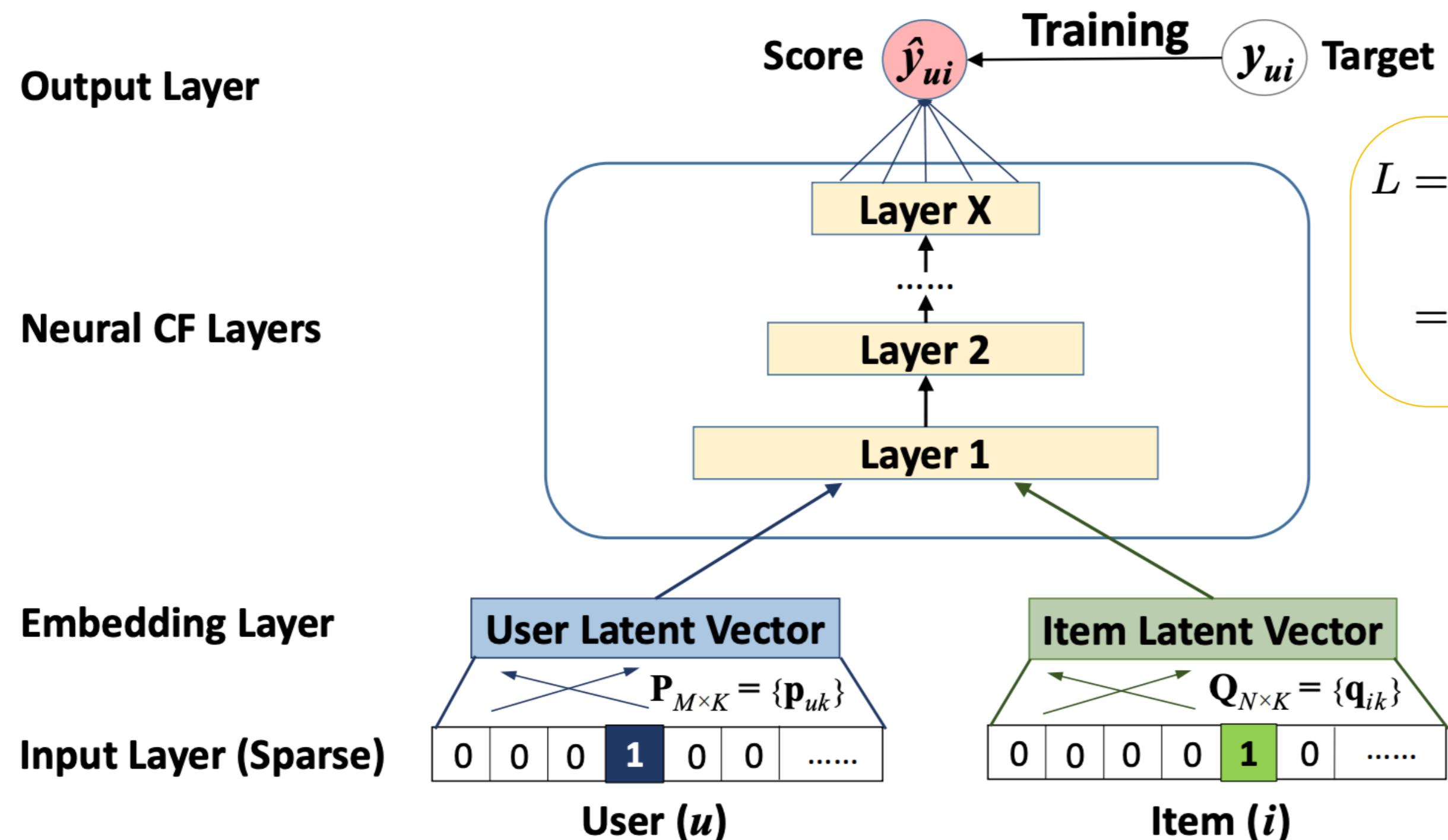
\*[https://making.lyst.com/lightfm/docs/examples/warp\\_loss.html](https://making.lyst.com/lightfm/docs/examples/warp_loss.html)

# Negative sampling

- Случайные айтемы без взаимодействия (Uniform Negative Sampling)
- Negative mining
- Hierarchy based (клик>просмотр>отсутствие наблюдения)

**Наконец-то первая нейронная  
сеть – NeurCF (NCF)**

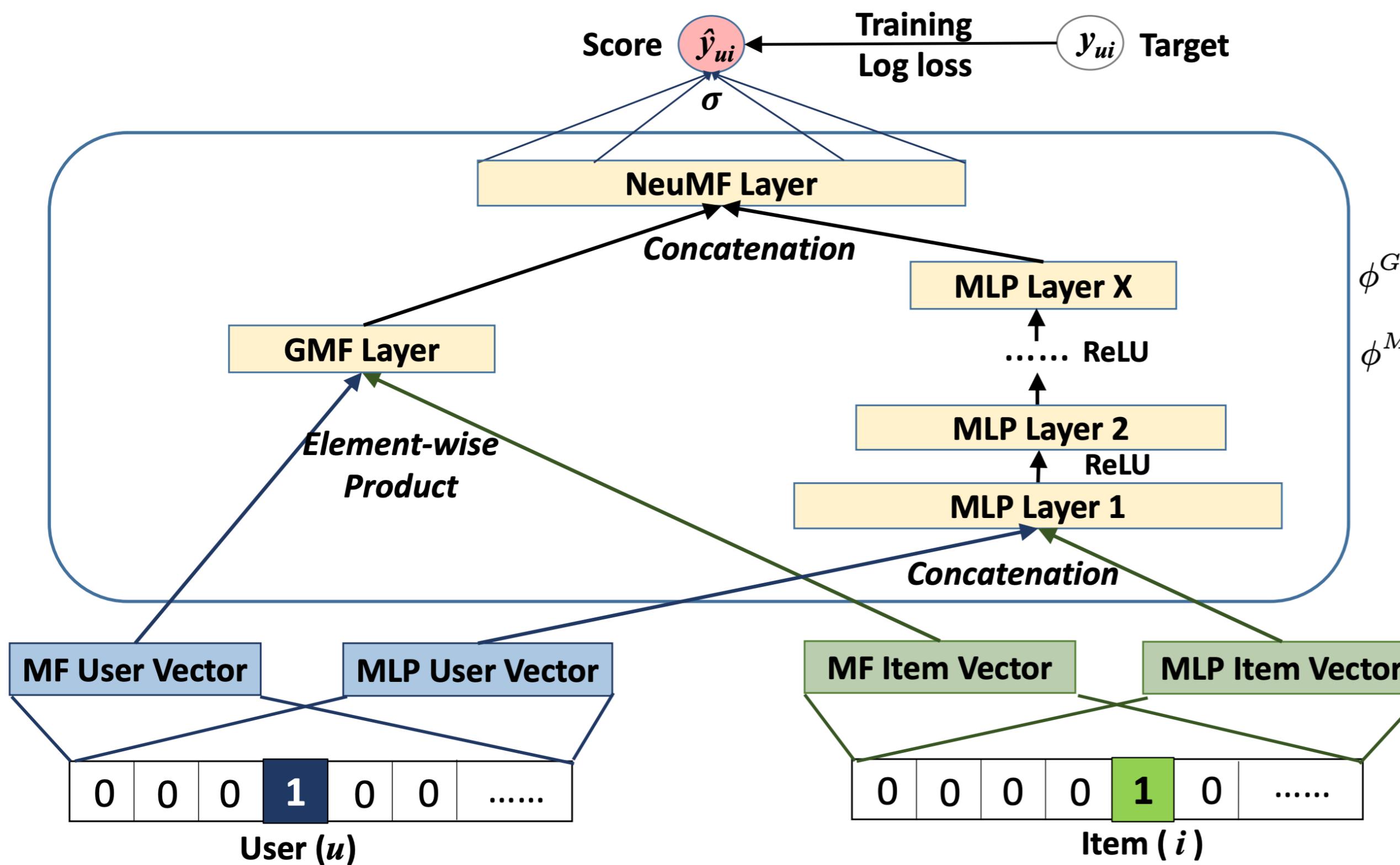
# Neural CF (NCF) 2017



$$\begin{aligned}
 L &= - \sum_{(u,i) \in \mathcal{Y}} \log \hat{y}_{ui} - \sum_{(u,j) \in \mathcal{Y}^-} \log(1 - \hat{y}_{uj}) \\
 &= - \sum_{(u,i) \in \mathcal{Y} \cup \mathcal{Y}^-} y_{ui} \log \hat{y}_{ui} + (1 - y_{ui}) \log(1 - \hat{y}_{ui}).
 \end{aligned}$$

$$\begin{aligned}
 \mathbf{z}_1 &= \phi_1(\mathbf{p}_u, \mathbf{q}_i) = \begin{bmatrix} \mathbf{p}_u \\ \mathbf{q}_i \end{bmatrix}, \\
 \phi_2(\mathbf{z}_1) &= a_2(\mathbf{W}_2^T \mathbf{z}_1 + \mathbf{b}_2), \\
 &\dots \\
 \phi_L(\mathbf{z}_{L-1}) &= a_L(\mathbf{W}_L^T \mathbf{z}_{L-1} + \mathbf{b}_L), \\
 \hat{y}_{ui} &= \sigma(\mathbf{h}^T \phi_L(\mathbf{z}_{L-1})),
 \end{aligned}$$

# Neural CF (NCF) 2017

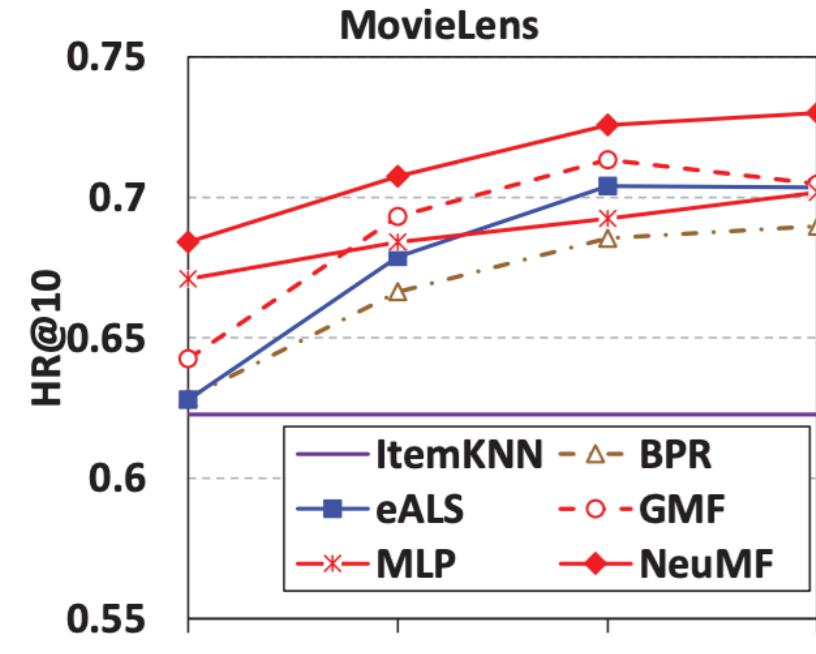


$$\phi^{GMF} = \mathbf{p}_u^G \odot \mathbf{q}_i^G,$$

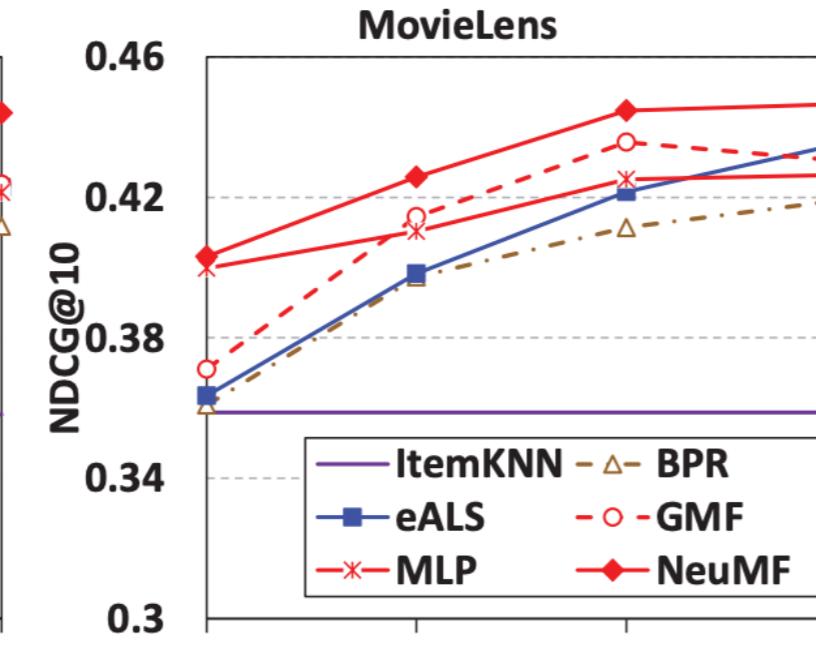
$$\phi^{MLP} = a_L(\mathbf{W}_L^T(a_{L-1}(\dots a_2(\mathbf{W}_2^T [\mathbf{p}_u^M \mathbf{q}_i^M] + \mathbf{b}_2)\dots)) + \mathbf{b}_L),$$

$$\hat{y}_{ui} = \sigma(\mathbf{h}^T [\phi^{GMF} \phi^{MLP}]),$$

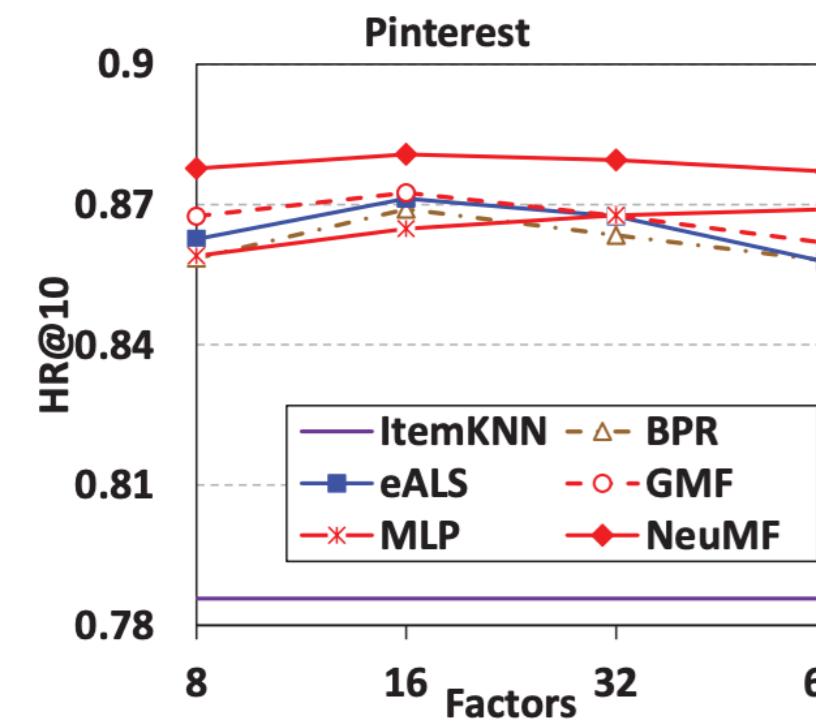
# Neural CF (NCF) 2017



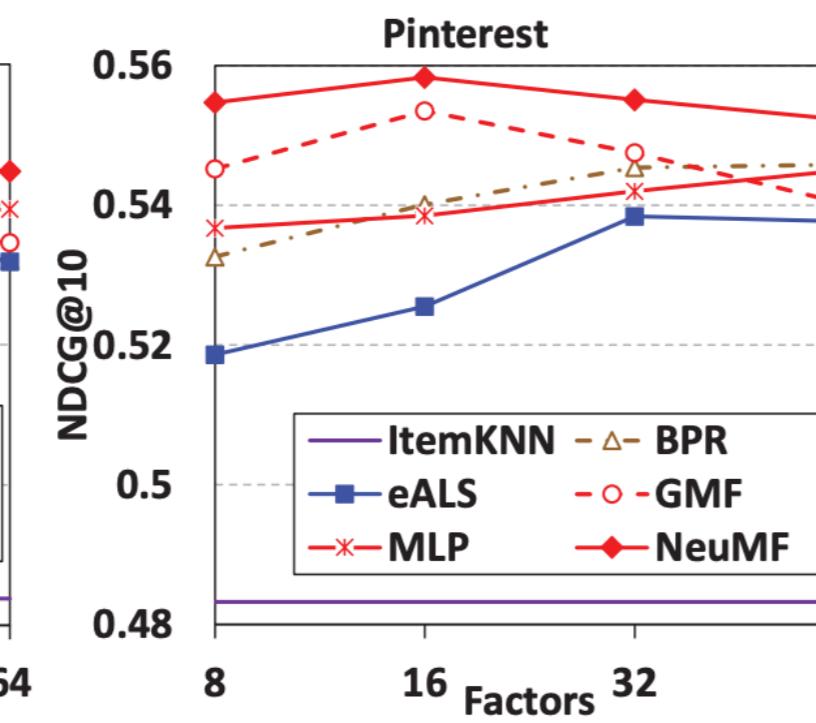
(a) MovieLens — HR@10



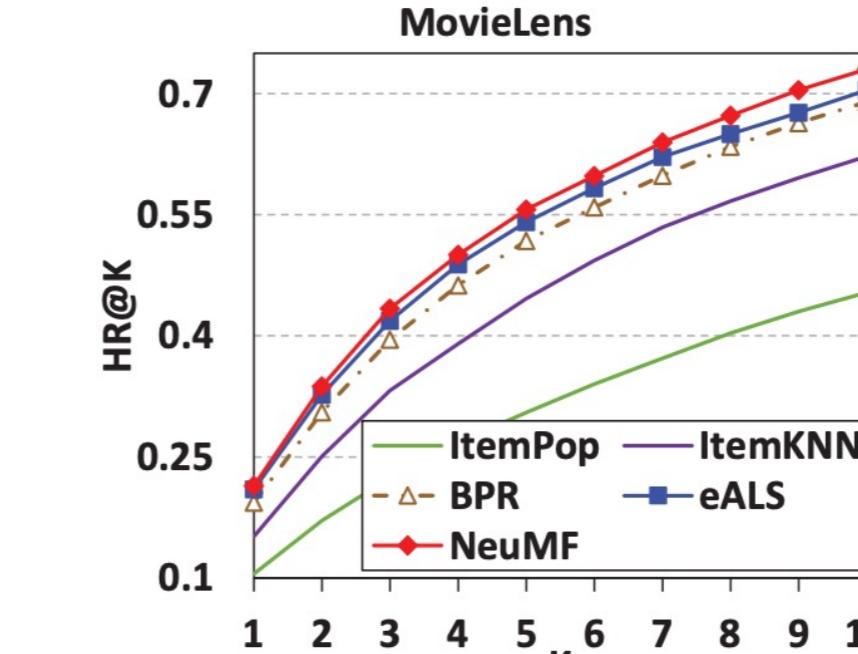
(b) MovieLens — NDCG@10



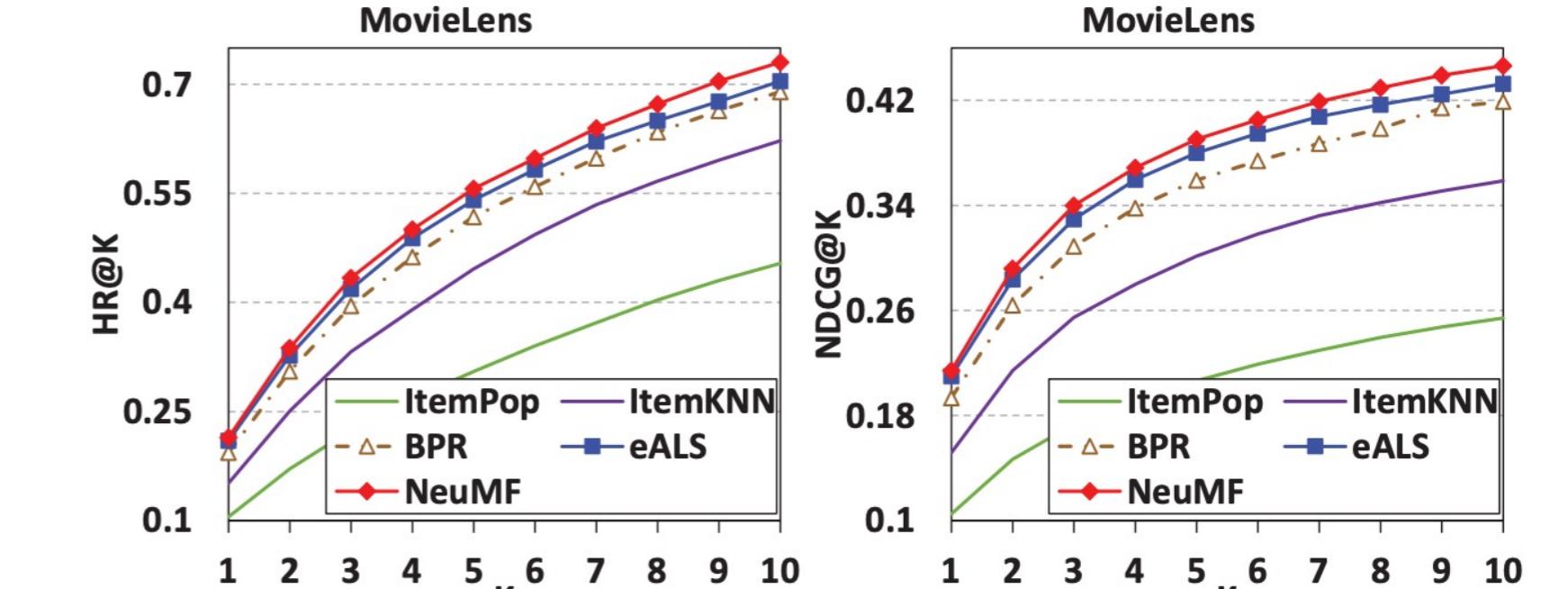
(c) Pinterest — HR@10



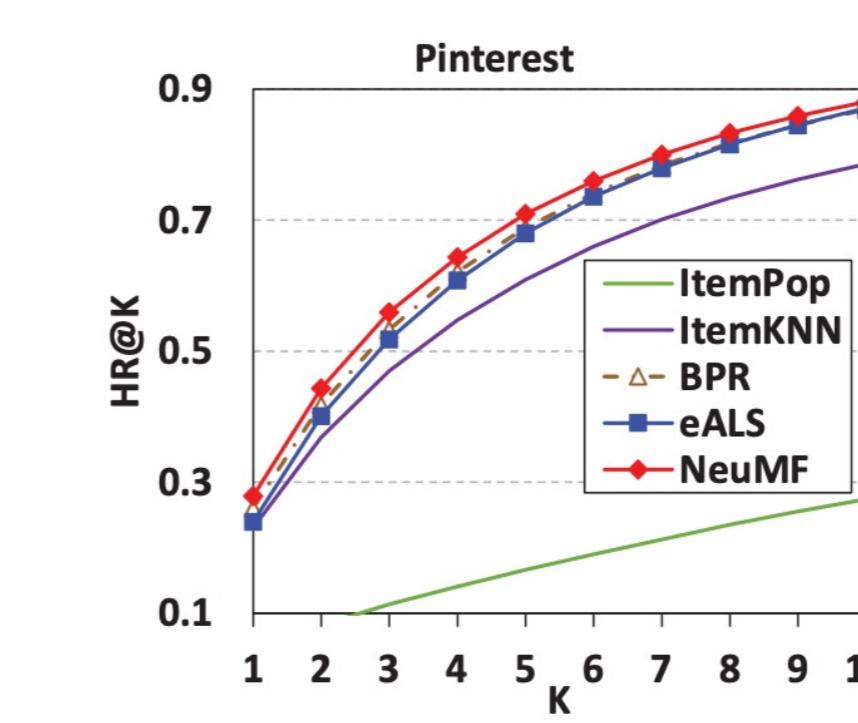
(d) Pinterest — NDCG@10



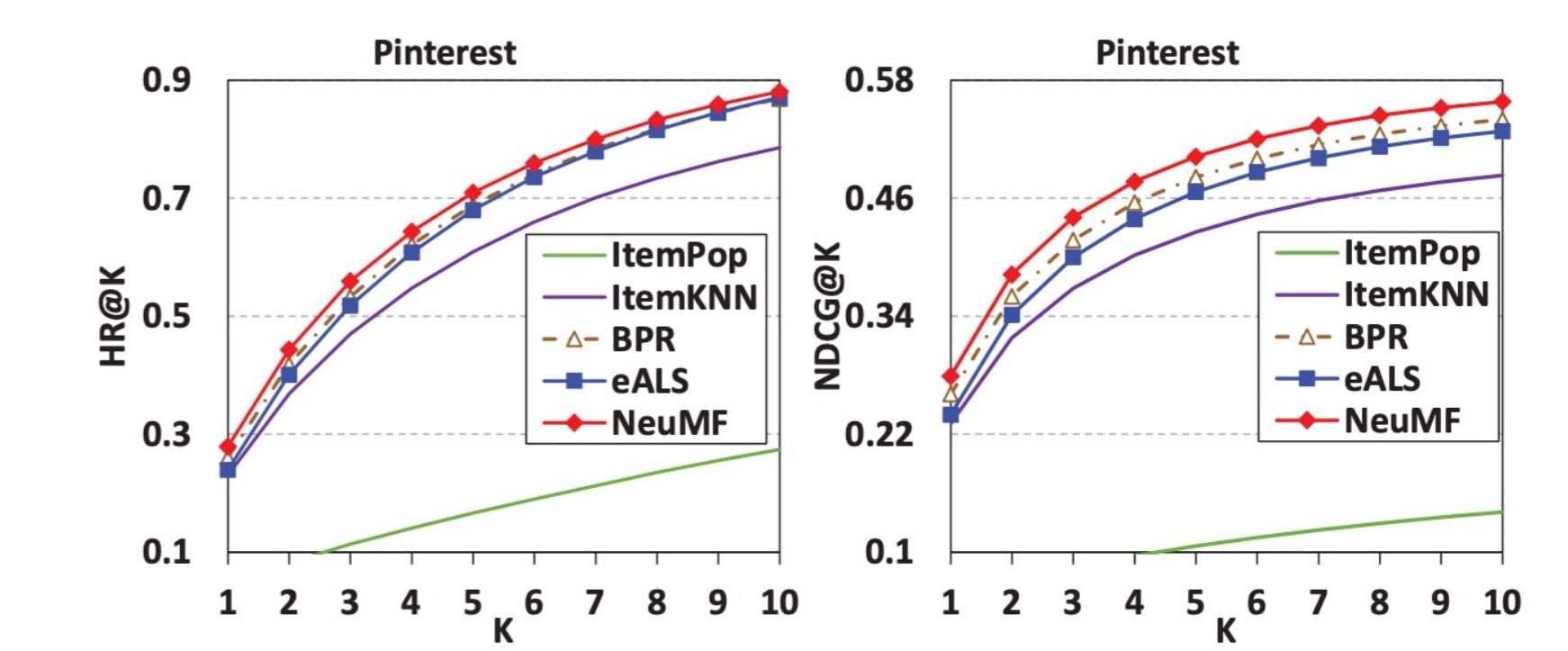
(a) MovieLens — HR@K



(b) MovieLens — NDCG@K



(c) Pinterest — HR@K



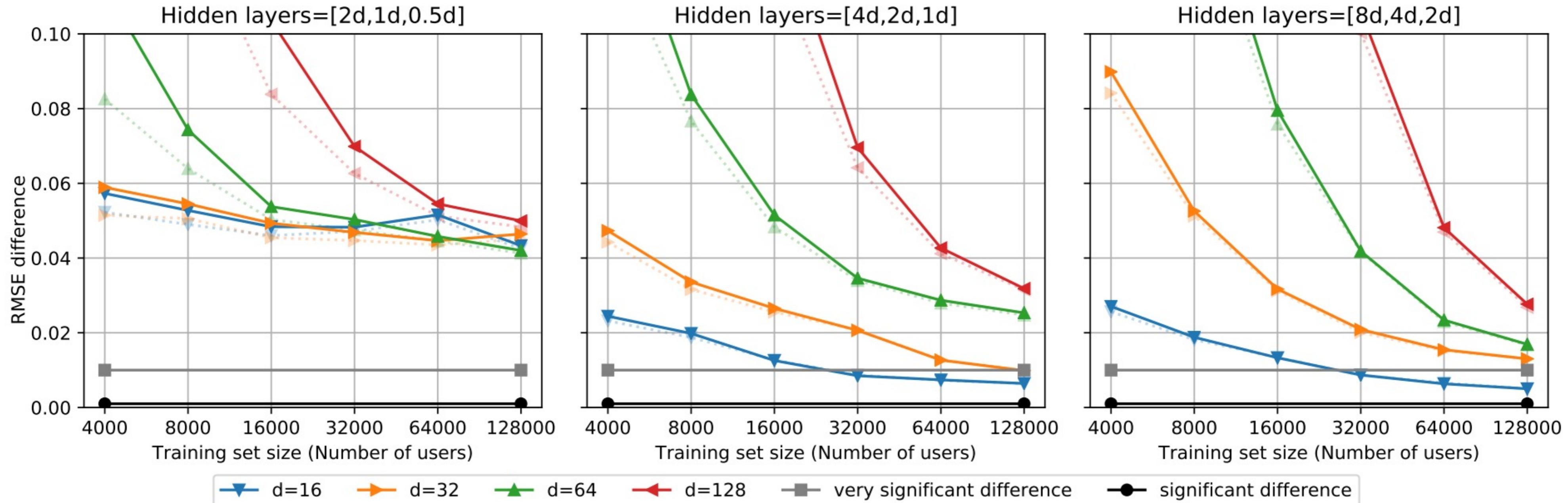
(d) Pinterest — NDCG@K

# LEARNING A DOT PRODUCT WITH MLP IS HARD

Method	Movielens		Pinterest		Result from
	HR@10	NDCG@10	HR@10	NDCG@10	
Popularity	0.4535	0.2543	0.2740	0.1409	[8]
SLIM [25, 30]	<u>0.7162</u>	<u>0.4468</u>	0.8679	<u>0.5601</u>	[8]
iALS [20]	0.7111	0.4383	0.8762	0.5590	[8]
NeuMF (MLP+GMF) [17]	0.7093	0.4349	<u>0.8777</u>	0.5576	[8]
Matrix Factorization	<b>0.7294</b>	<b>0.4523</b>	<b>0.8895</b>	<b>0.5794</b>	Fig. 2

Rendle S, Krichene W, Zhang L, Anderson J. Neural collaborative filtering vs. matrix factorization revisited. In Proceedings of the 14th ACM Conference on Recommender Systems 2020 Sep 22 (pp. 240-248).

# LEARNING A DOT PRODUCT WITH MLP IS HARD



Rendle S, Krichene W, Zhang L, Anderson J. Neural collaborative filtering vs. matrix factorization revisited. In Proceedings of the 14th ACM Conference on Recommender Systems 2020 Sep 22 (pp. 240-248).

Спасибо за  
внимание