
Курс: Рекомендательные системы

Занятие 5:

Контентные и контекстные подходы

Красильников Денис
24.02.24

Контент и контекст

Идея: Дать дополнительную полезную информацию по взаимодействиям пользователя, при помощи описания объекта, пользователя или условий, при которых было совершено взаимодействие.



80%

of content consumed on
Netflix is due to
recommendations.



60%

of video clicks on
Youtube's homepage
are attributed to
recommendations



35%

of its revenue is generated
by its recommendation
engine

Контент и контекст



Идея: Дать дополнительную полезную информацию по взаимодействиям пользователя, при помощи описания объекта, пользователя или условий, при которых было совершено взаимодействие.

Контент –

Контекст –

Контент и контекст



Идея: Дать дополнительную полезную информацию по взаимодействиям пользователя, при помощи описания объекта, пользователя или условий, при которых было совершено взаимодействие.

Контент – описательное свойство объекта/пользователя, которое, как правило, не изменяется во времени или делает это очень редко.

Контекст –

Контент и контекст



Идея: Дать дополнительную полезную информацию по взаимодействиям пользователя, при помощи описания объекта, пользователя или условий, при которых было совершено взаимодействие.

Контент – описательное свойство объекта/пользователя, которое, как правило, не изменяется во времени или делает это очень редко.

Контекст – внешние условия или быстро меняющиеся признаки.

Контент и контекст



Контекст:

- Время
- Геолокация
- Девайс получения рекомендаций
- Количество денег на счету
- Поисковый запрос недавний
- Межличностные отношения
- Внешние факторы
- Погода на улице

Контент:

- Описание
- Категории товара
- Пол пользователя
- Имя пользователя
- Режиссер фильма
- Жанр фильма
- Актер главной роли
- Знак зодиака пользователя

Как добавить контекст/контакт в модели?

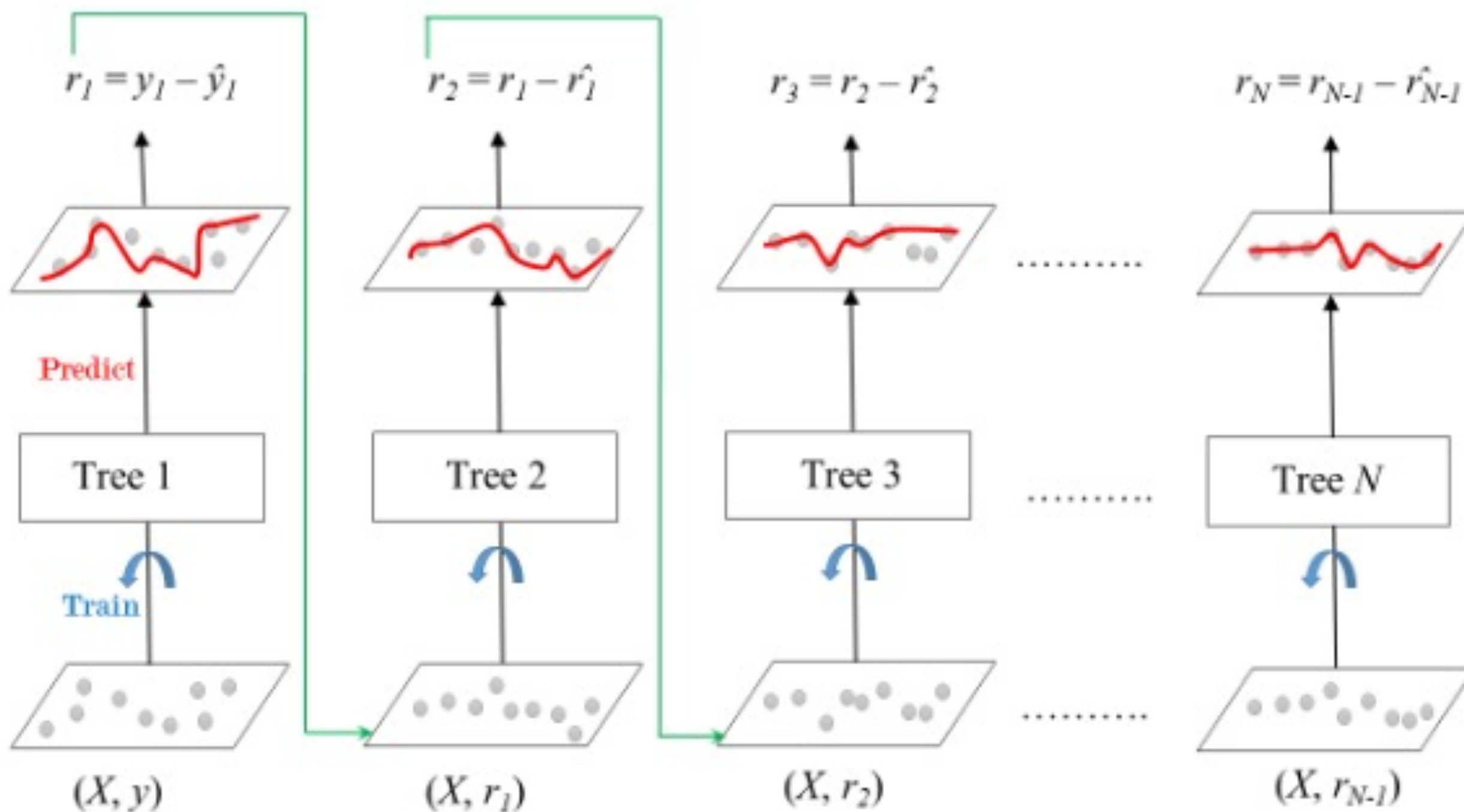


Gradient boosting

 LightGBM

 CatBoost

dmlc
XGBoost



Gradient boosting



Logloss

$$-\sum_{i=1}^N w_i (c_i \log(p_i) + (1 - c_i) \log(1 - p_i))$$

$$\sum_{i=1}^N w_i$$

PairLogit

$$-\frac{\sum_{p,n \in \text{Pairs}} w_{pn} \left(\log\left(\frac{1}{1 + e^{-(a_p - a_n)}}\right) \right)}{\sum_{p,n \in \text{Pairs}} w_{pn}}$$

YetiRank

$$\mathbb{L} = -\sum_{(i,j)} w_{ij} \log \frac{e^{x_i}}{e^{x_i} + e^{x_j}},$$

Deep Structured Semantic Model



Learning Deep Structured Semantic Models for Web Search using Clickthrough Data

Po-Sen Huang

University of Illinois at Urbana-Champaign
405 N Mathews Ave. Urbana, IL 61801 USA
huang146@illinois.edu

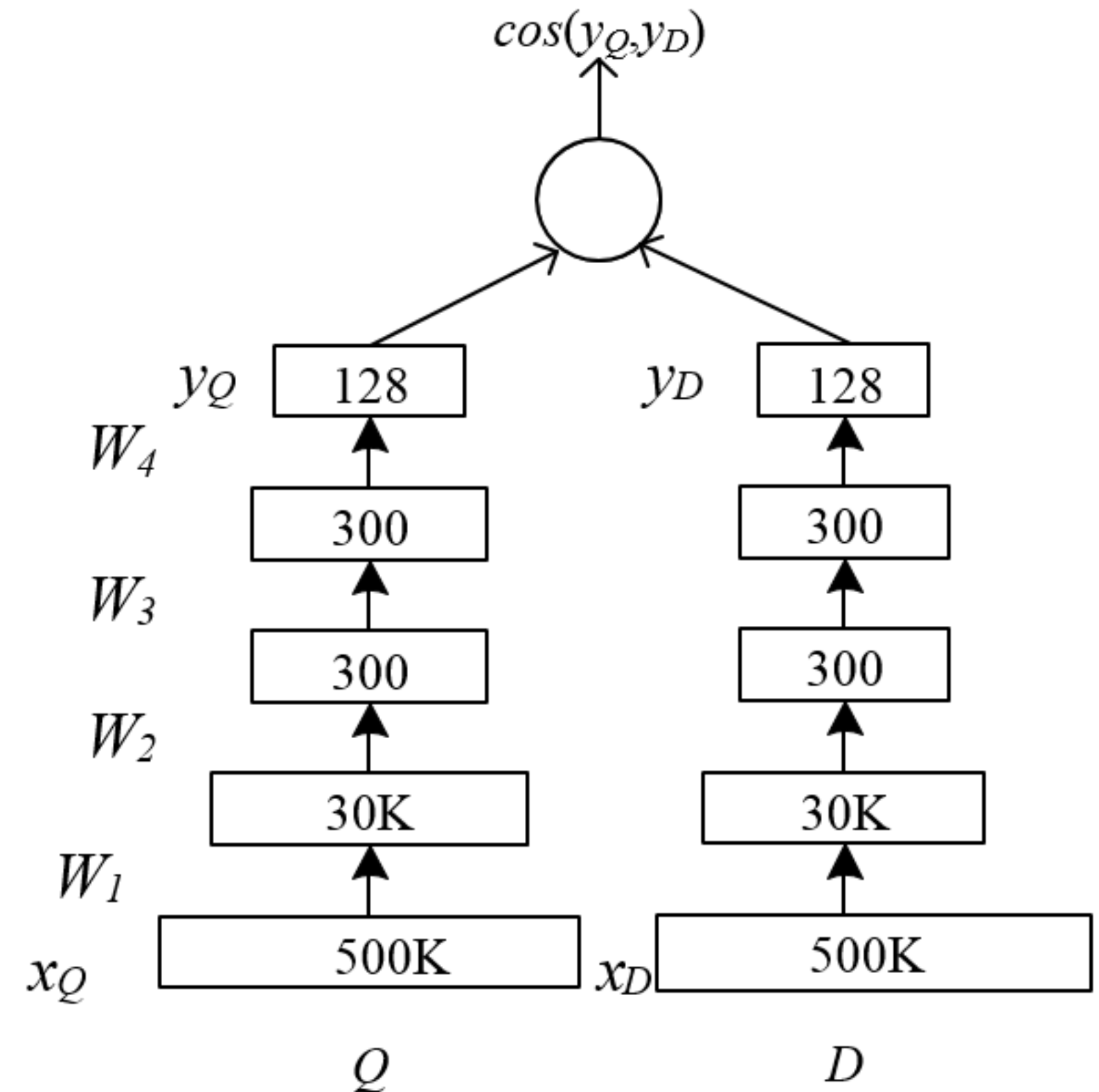
Xiaodong He, Jianfeng Gao, Li Deng,

Alex Acero, Larry Heck

Microsoft Research, Redmond, WA 98052 USA
{xiaohe, jfgao, deng, alexac, lheck}@microsoft.com

ABSTRACT

Latent semantic models, such as LSA, intend to map a query to its relevant documents at the semantic level where keyword-based matching often fails. In this study we strive to develop a series of new latent semantic models with a deep structure that project queries and documents into a common low-dimensional space where the relevance of a document given a query is readily computed as the distance between them. The proposed deep structured semantic models are discriminatively trained by maximizing the conditional likelihood of the clicked documents given a query using the clickthrough data. To make our models applicable to large-scale Web search applications, we also use a technique called word hashing, which is shown to effectively scale up our semantic models to handle large vocabularies which are common in such tasks. The new models are evaluated on a Web document ranking task using a real-world data set. Results show that our best model significantly outperforms other latent semantic models, which were considered state-of-the-art in the performance prior to the work presented in this paper.



Huang PS, He X, Gao J, Deng L, Acero A, Heck L. Learning deep structured semantic models for web search using clickthrough data. In Proceedings of the 22nd ACM international conference on Information & Knowledge Management 2013 Oct 27 (pp. 2333-2338).

Deep Structured Semantic Model

Posterior probability
computed by softmax

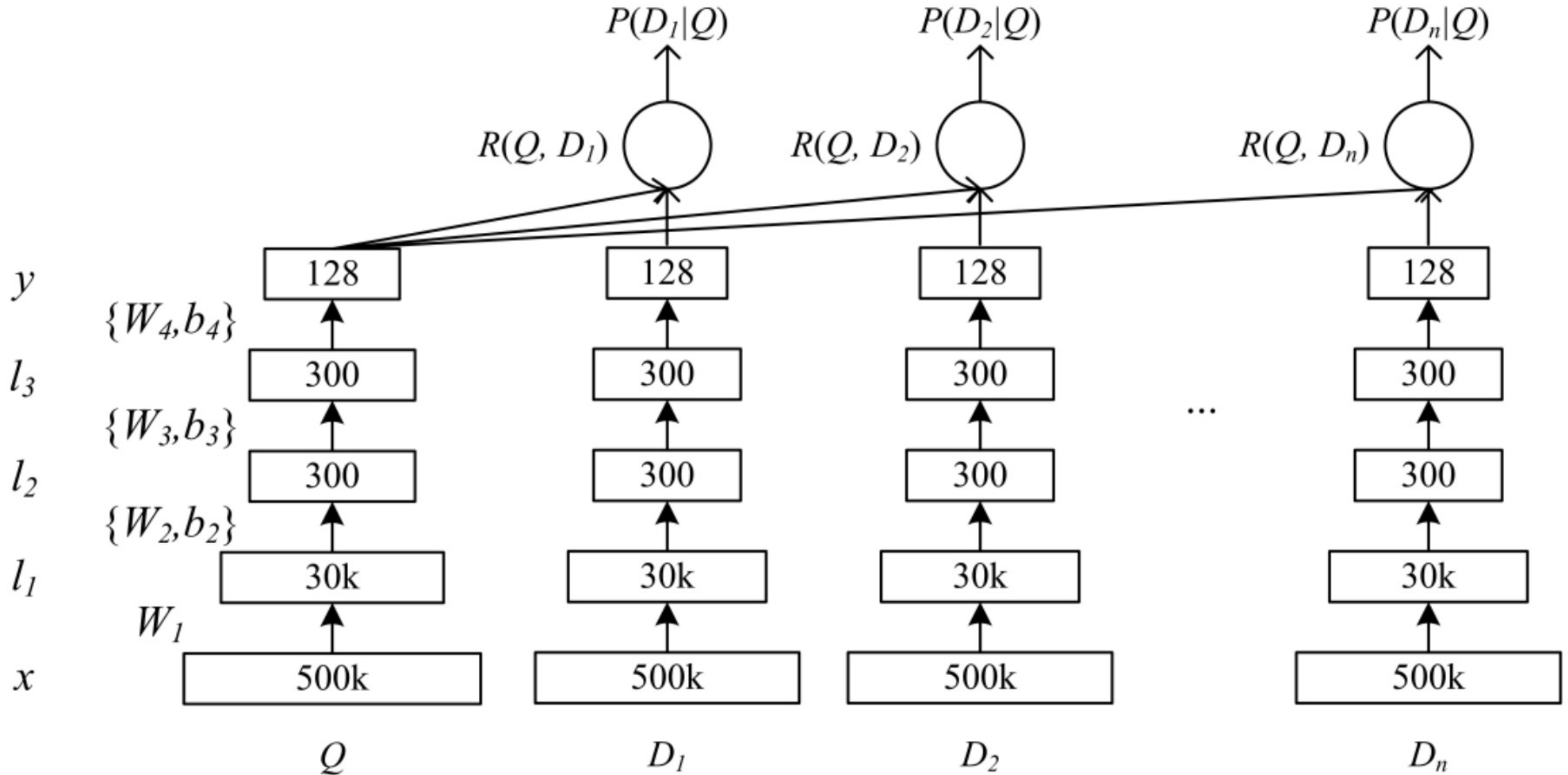
Relevance measured
by cosine similarity

Semantic feature

Multi-layer non-
linear projection

Word Hashing

Term Vector



Deep Structured Semantic Model



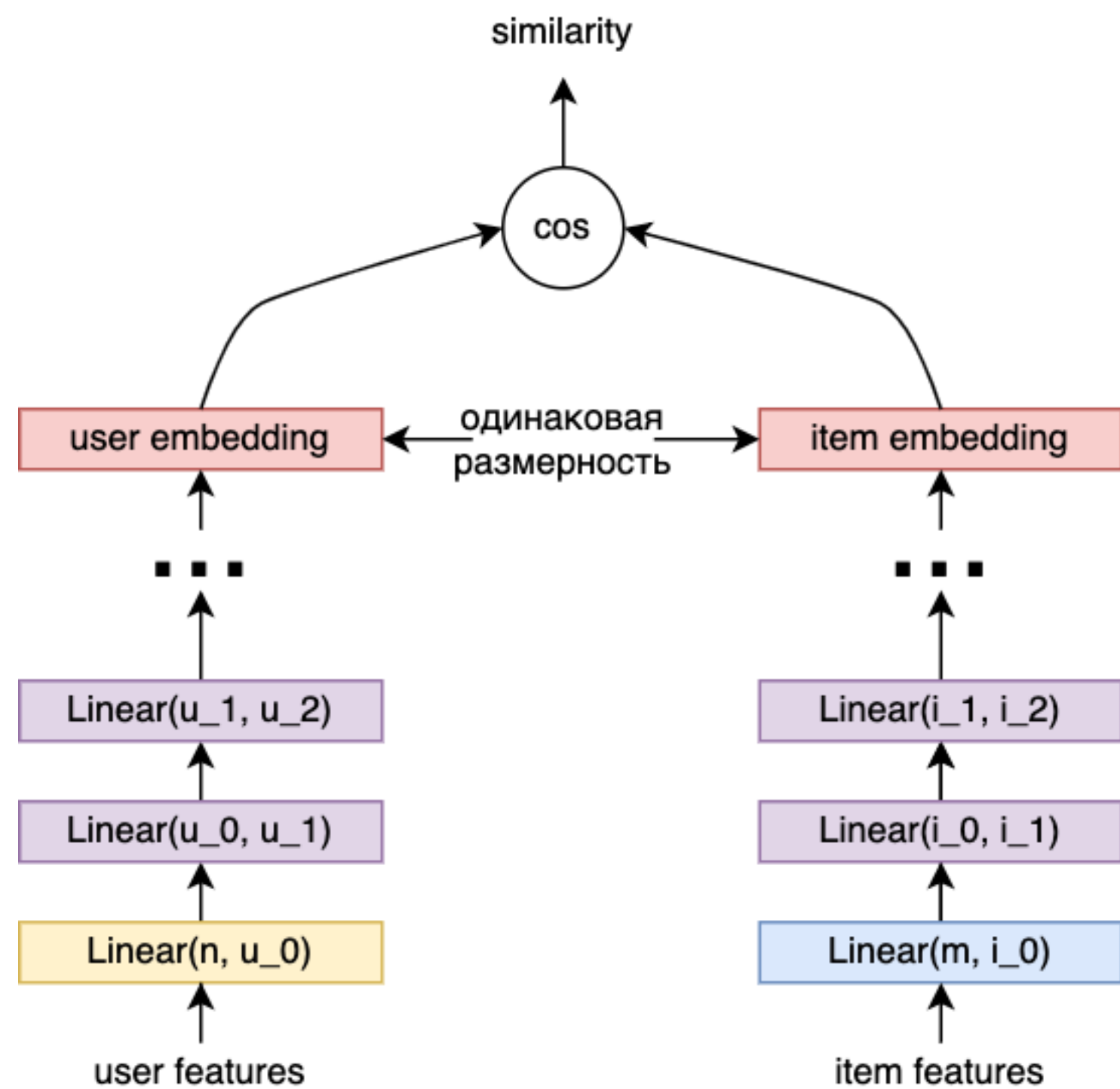
$$R(Q, D) = \text{cosine}(y_Q, y_D) = \frac{y_Q^T y_D}{\|y_Q\| \|y_D\|}$$

$$P(D|Q) = \frac{\exp(\gamma R(Q, D))}{\sum_{D' \in D} \exp(\gamma R(Q, D'))} \quad \gamma - \text{smoothing factor}$$

$$L(\Lambda) = -\log \prod_{(Q, D^+)} P(D^+|Q)$$

D^+ – позитивный пример
 D^- – негативные примеры
 $D = D^+ \cup D^-$

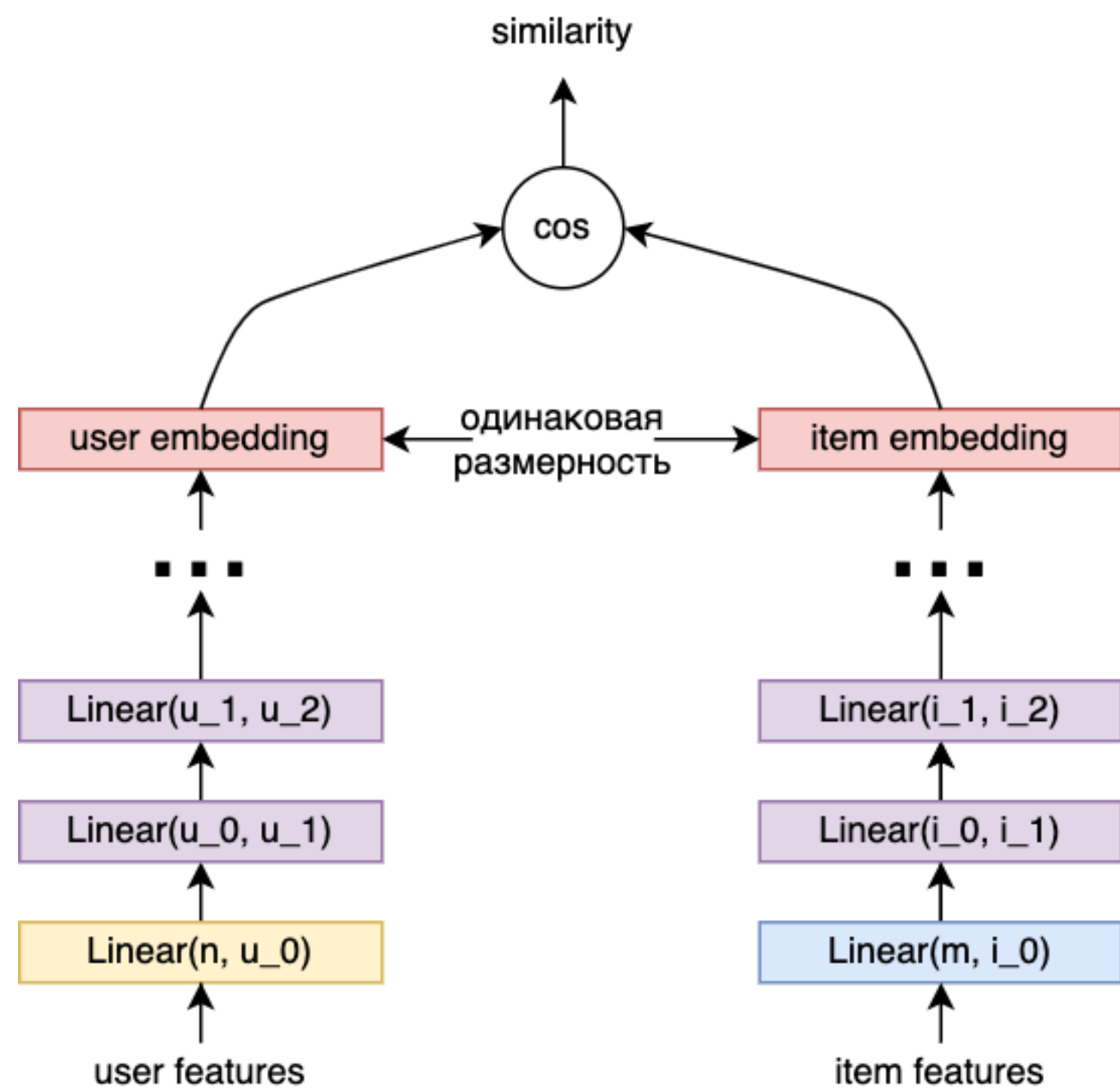
Deep Structured Semantic Model



user $item_0^-$ $item_1^-$ $item_2^-$ $item^+$ $item_3^-$

-0.0218

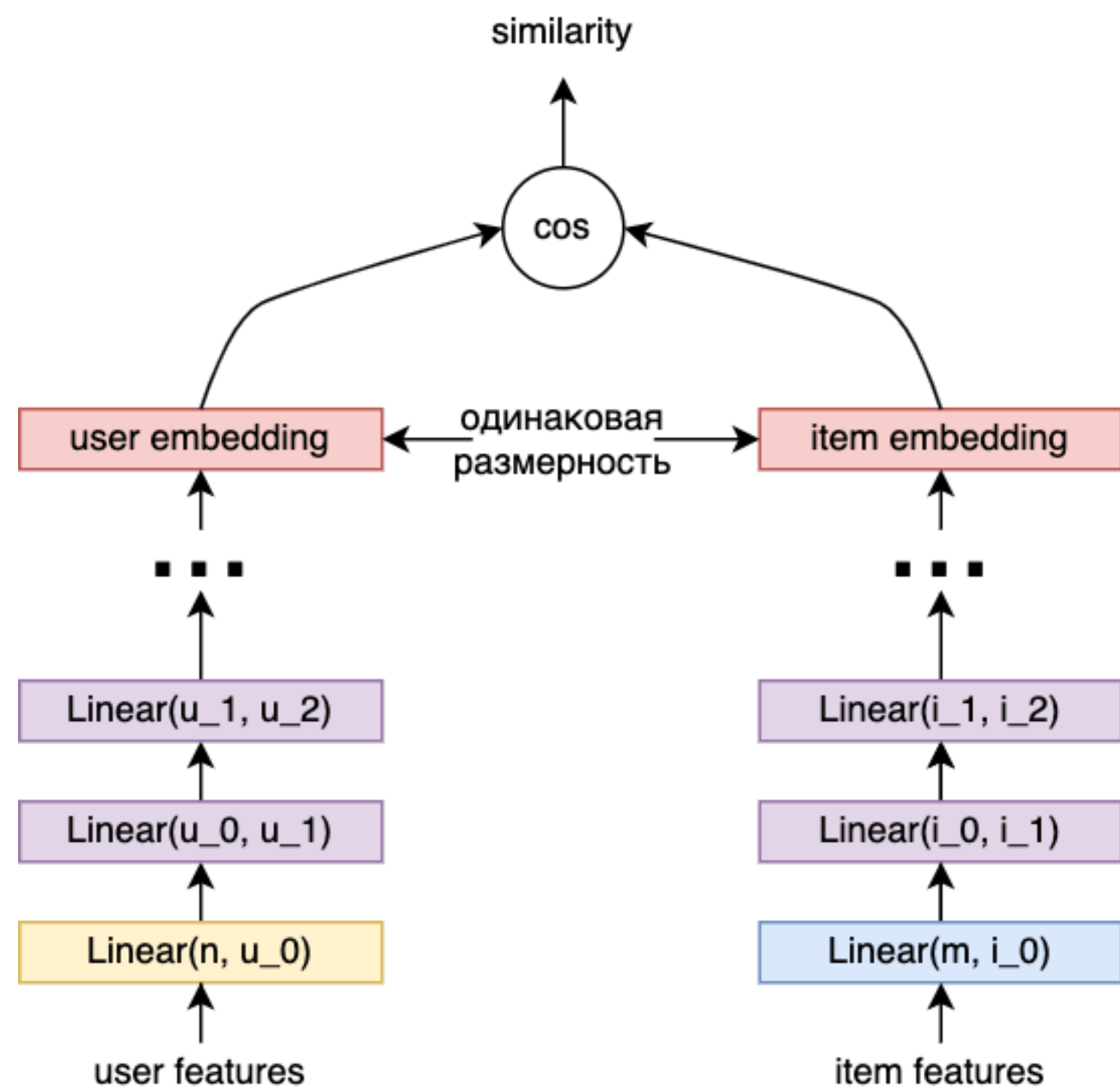
Deep Structured Semantic Model



user $item_0^-$ $item_1^-$ $item_2^-$ $item^+$ $item_3^-$

-0.0218 -0.4689

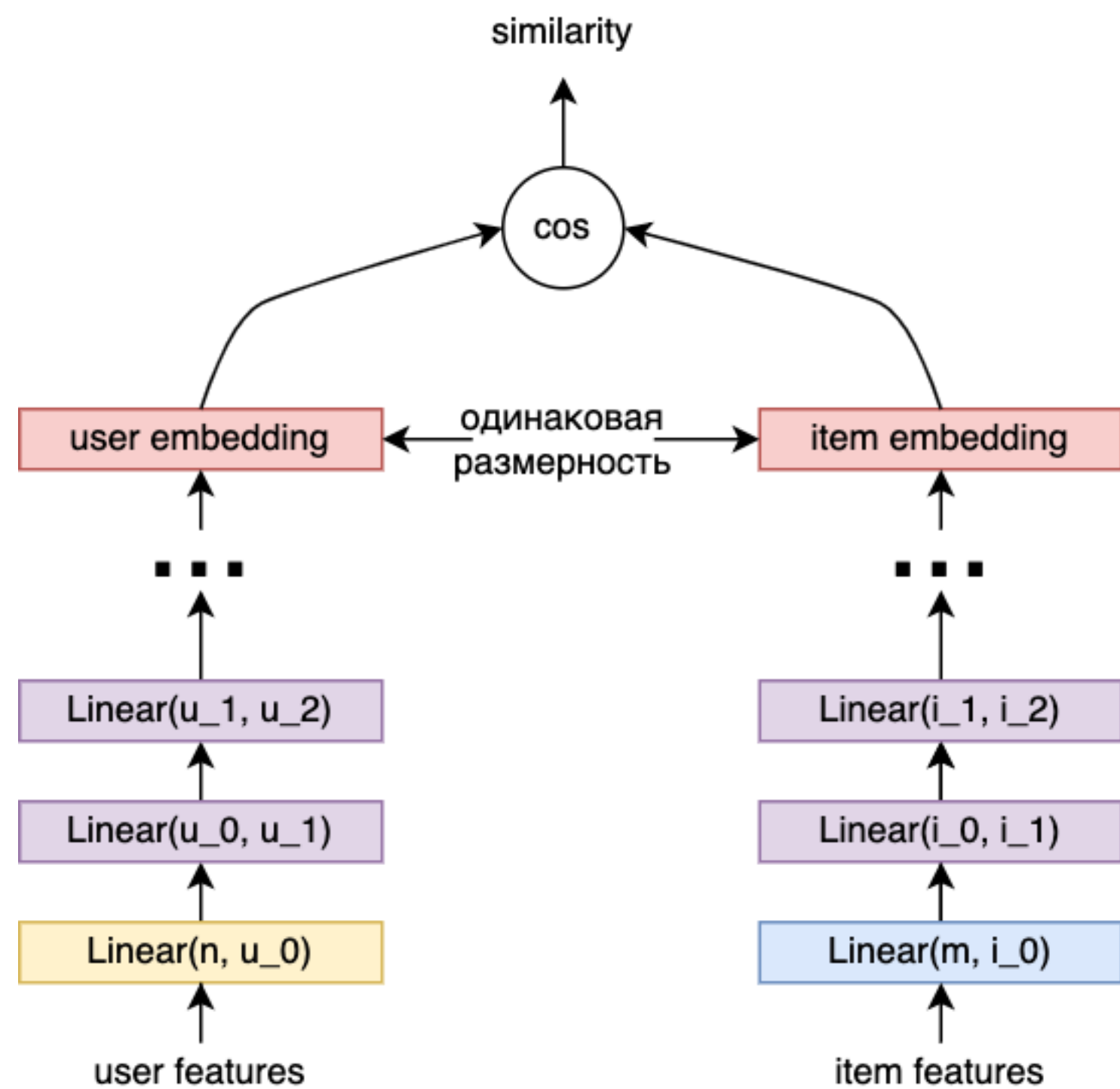
Deep Structured Semantic Model



user $item_0^-$ $item_1^-$ $item_2^-$ $item^+$ $item_3^-$

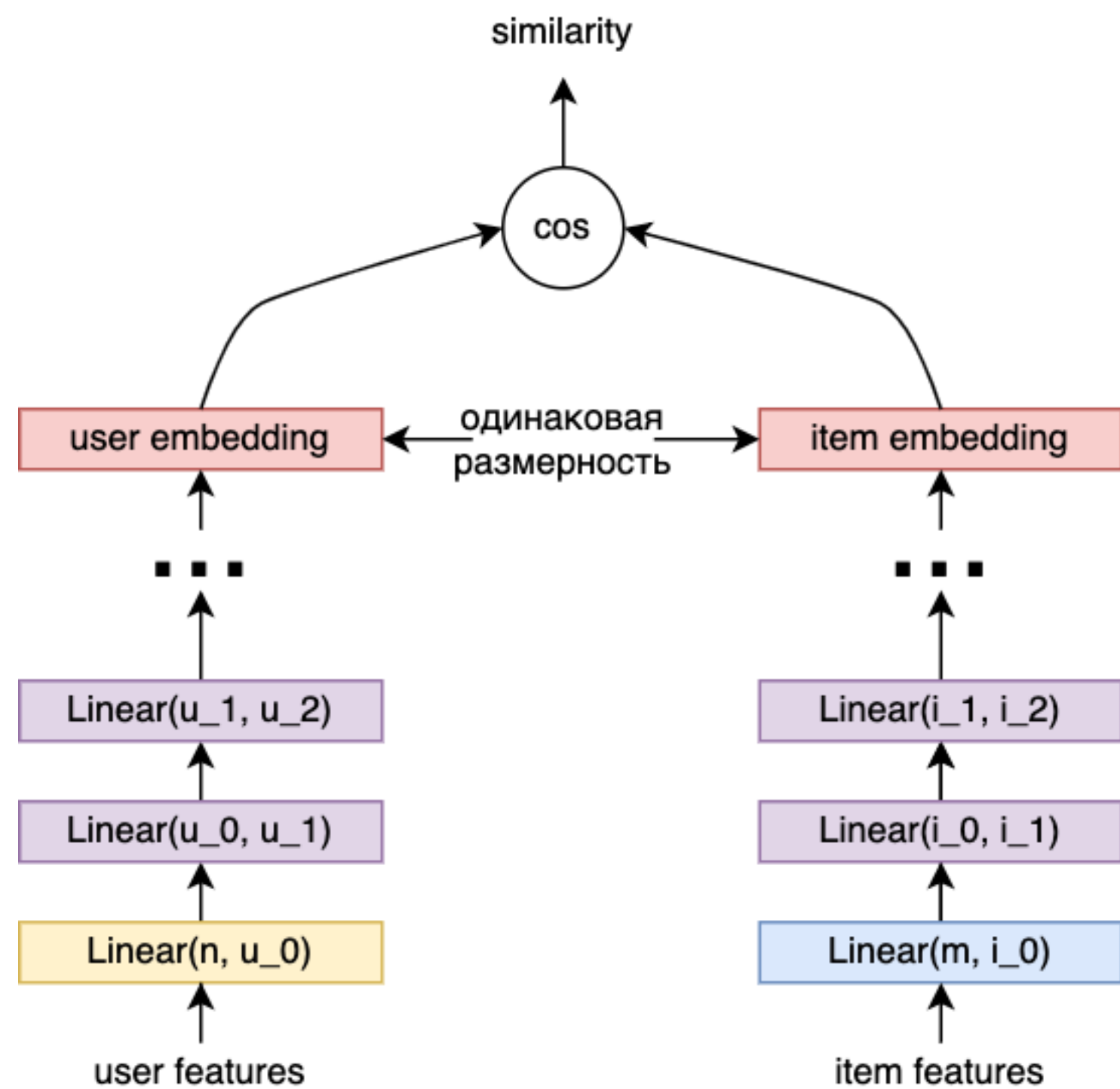
-0.0218 -0.4689 0.4136

Deep Structured Semantic Model



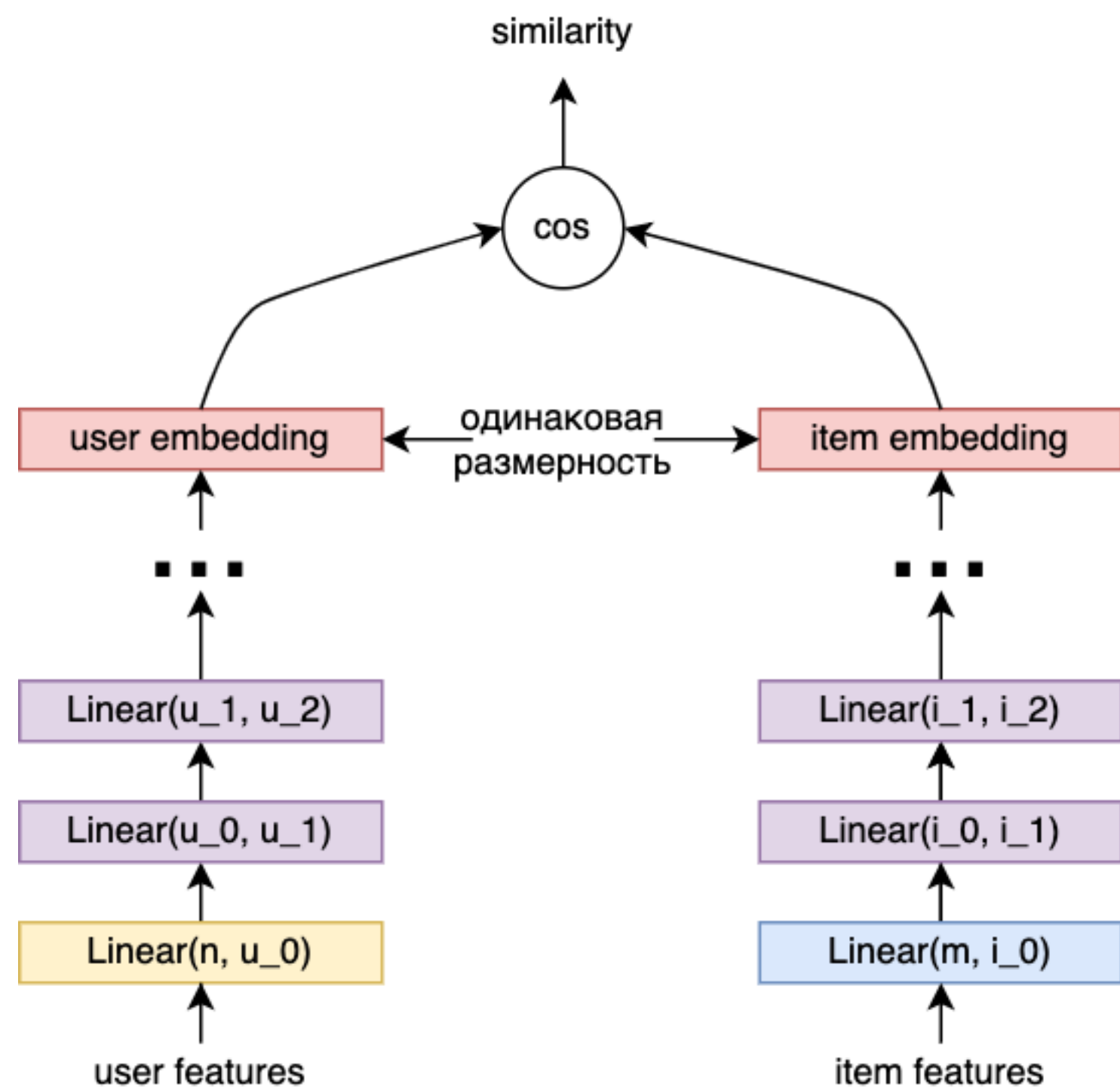
user	$item_0^-$	$item_1^-$	$item_2^-$	$item^+$	$item_3^-$
	-0.0218	-0.4689	0.4136	0.7126	

Deep Structured Semantic Model



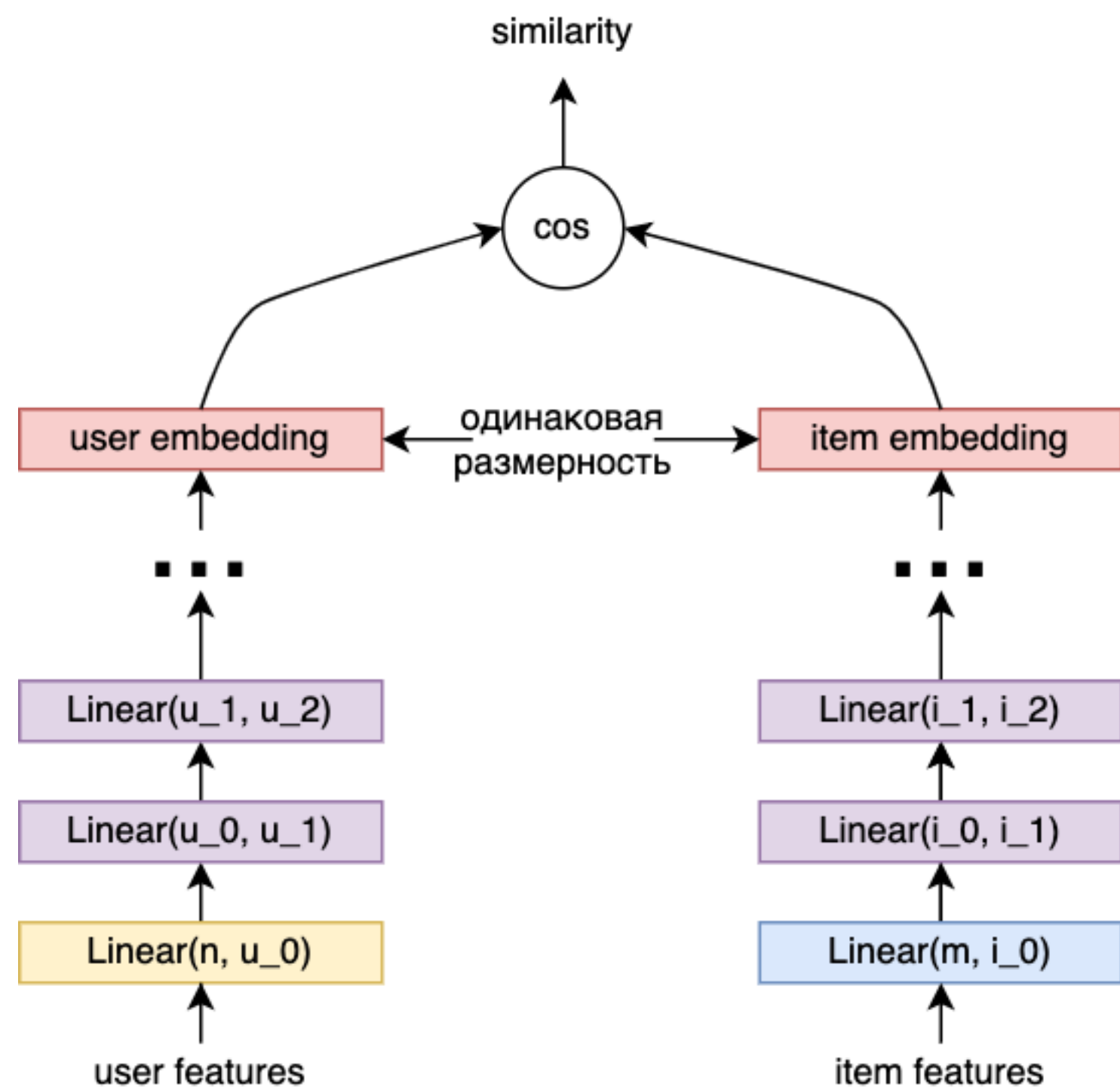
user	$item_0^-$	$item_1^-$	$item_2^-$	$item^+$	$item_3^-$
	-0.0218	-0.4689	0.4136	0.7126	-0.6863

Deep Structured Semantic Model



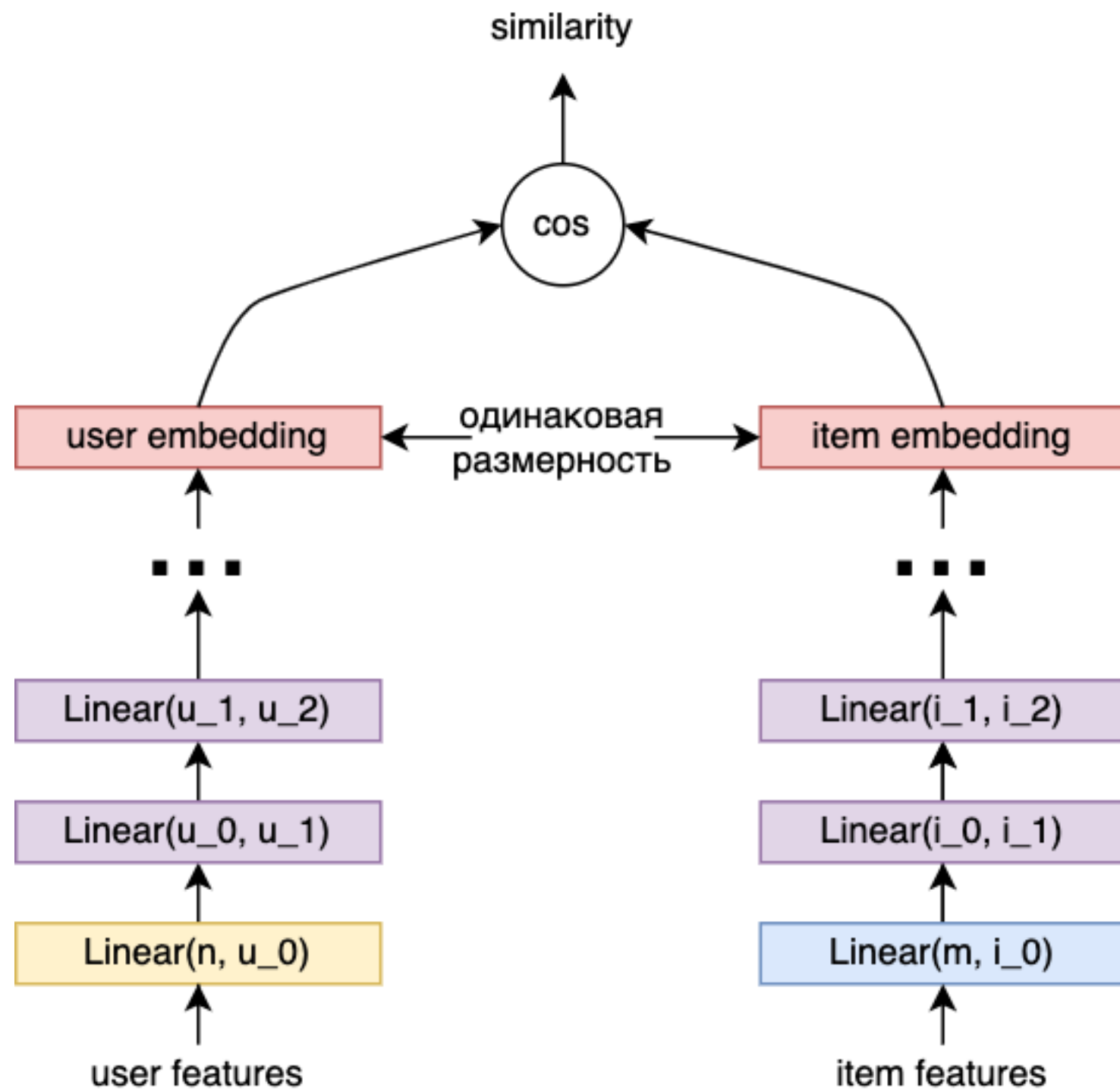
user	$item_0^-$	$item_1^-$	$item_2^-$	$item^+$	$item_3^-$
	-0.0218	-0.4689	0.4136	0.7126	-0.6863
	\downarrow				
	$Softmax(\gamma * x)$				
	\downarrow				
	0.1729	0.1106	0.2672	0.3604	0.0890

Deep Structured Semantic Model



user	$item_0^-$	$item_1^-$	$item_2^-$	$item^+$	$item_3^-$
	-0.0218	-0.4689	0.4136	0.7126	-0.6863
$Softmax(\gamma * x)$					
	0.1729	0.1106	0.2672	0.3604	0.0890

Deep Structured Semantic Model



user $item_0^-$ $item_1^-$ $item_2^-$ $item^+$ $item_3^-$

-0.0218 -0.4689 0.4136 0.7126 -0.6863

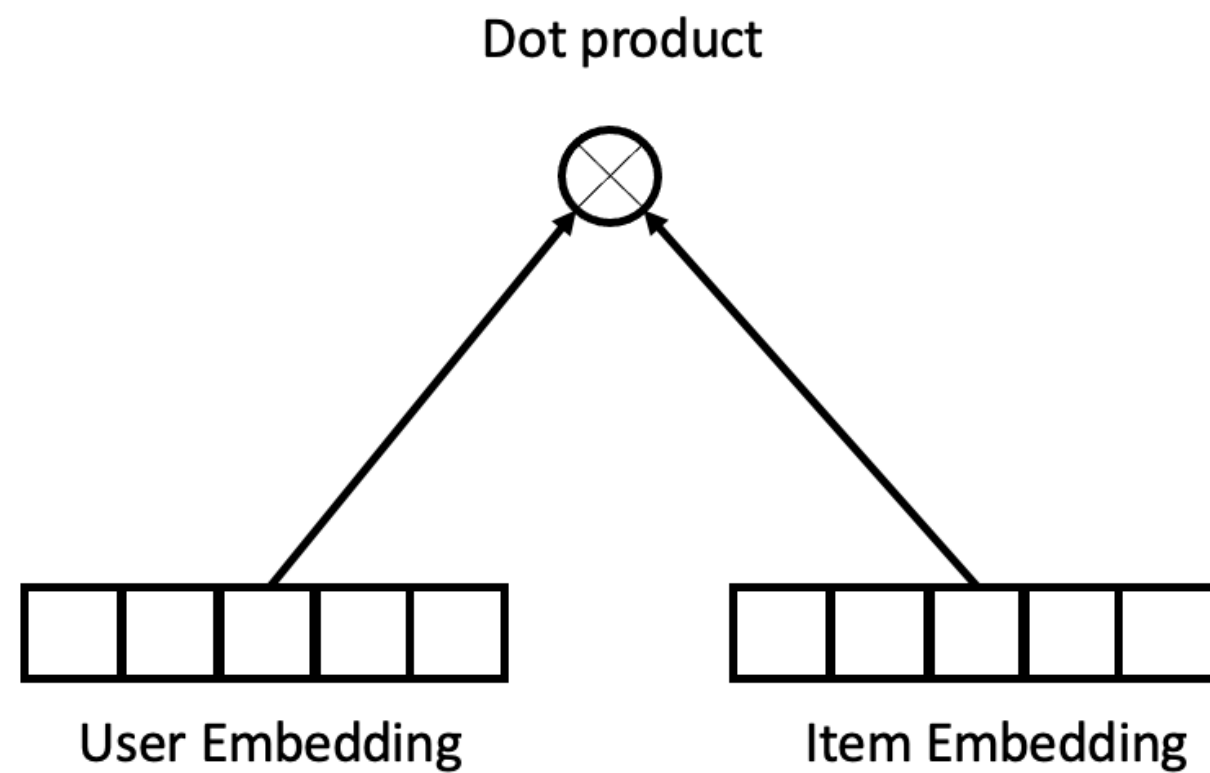
$\text{Softmax}(\gamma * x)$

0.1729 0.1106 0.2672 0.3604 0.0890

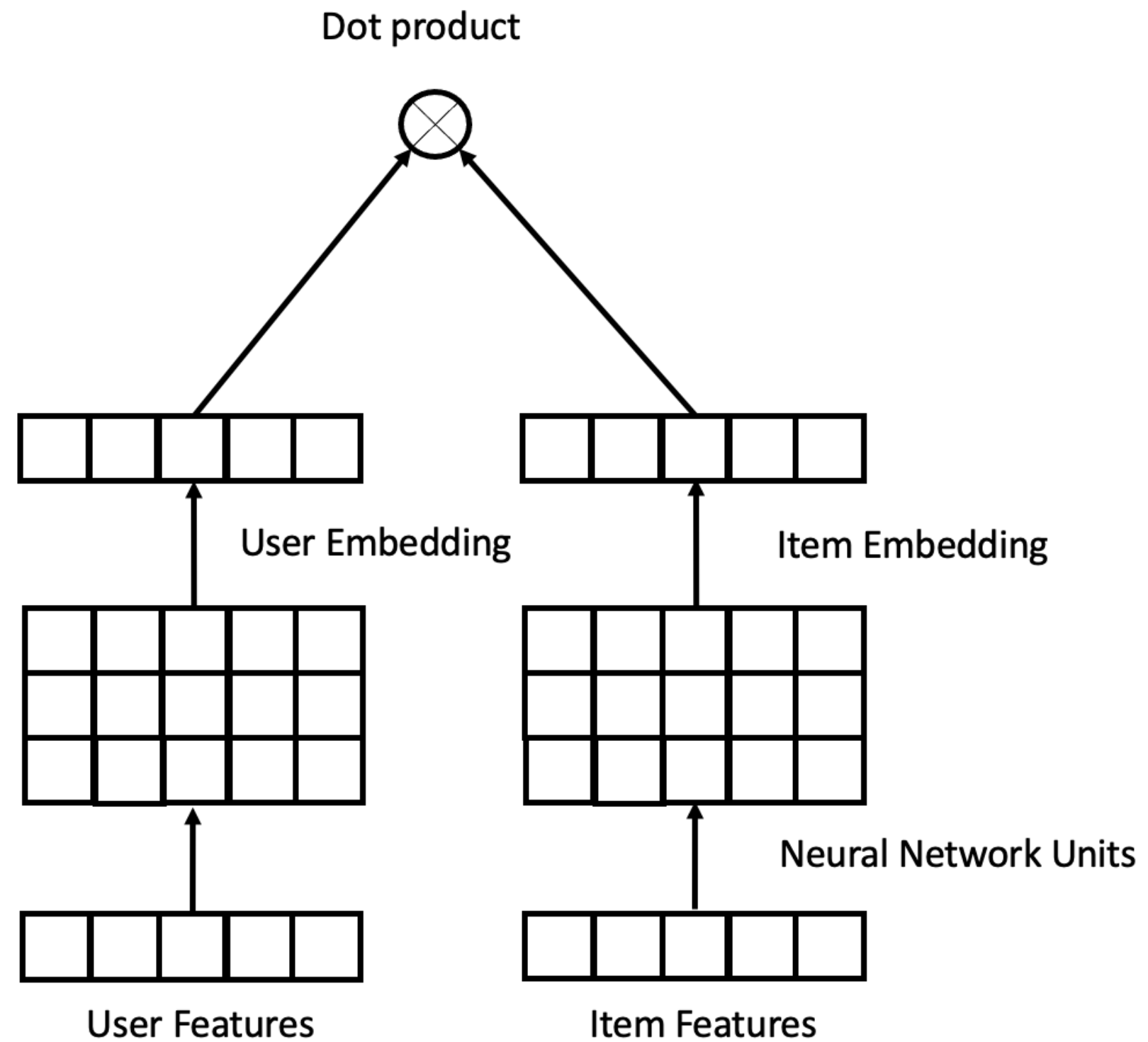
$$L = -\log(0.3604) = 1.02$$

Deep Structured Semantic Model

a) Matrix Factorization task



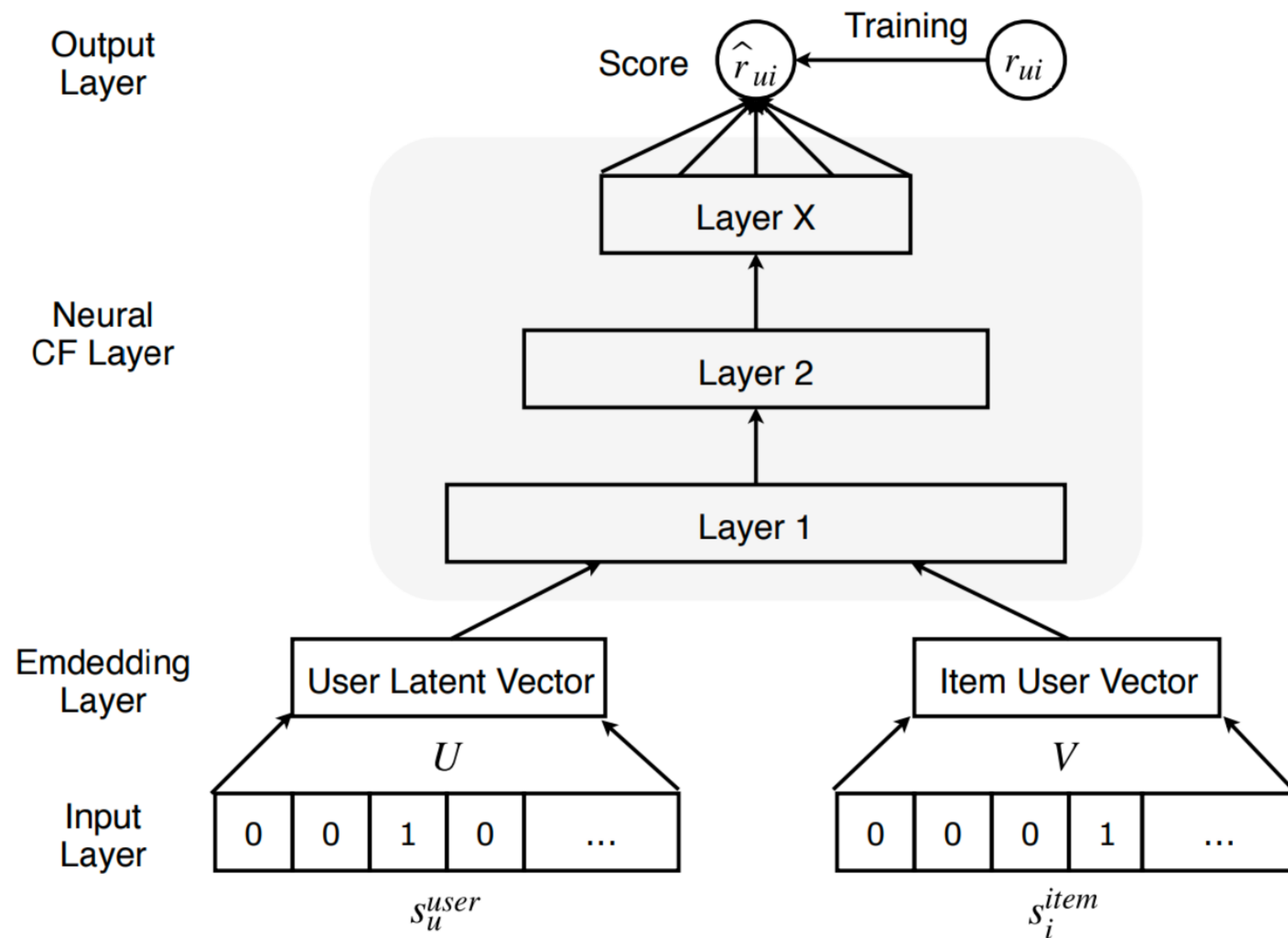
b) DSSM task



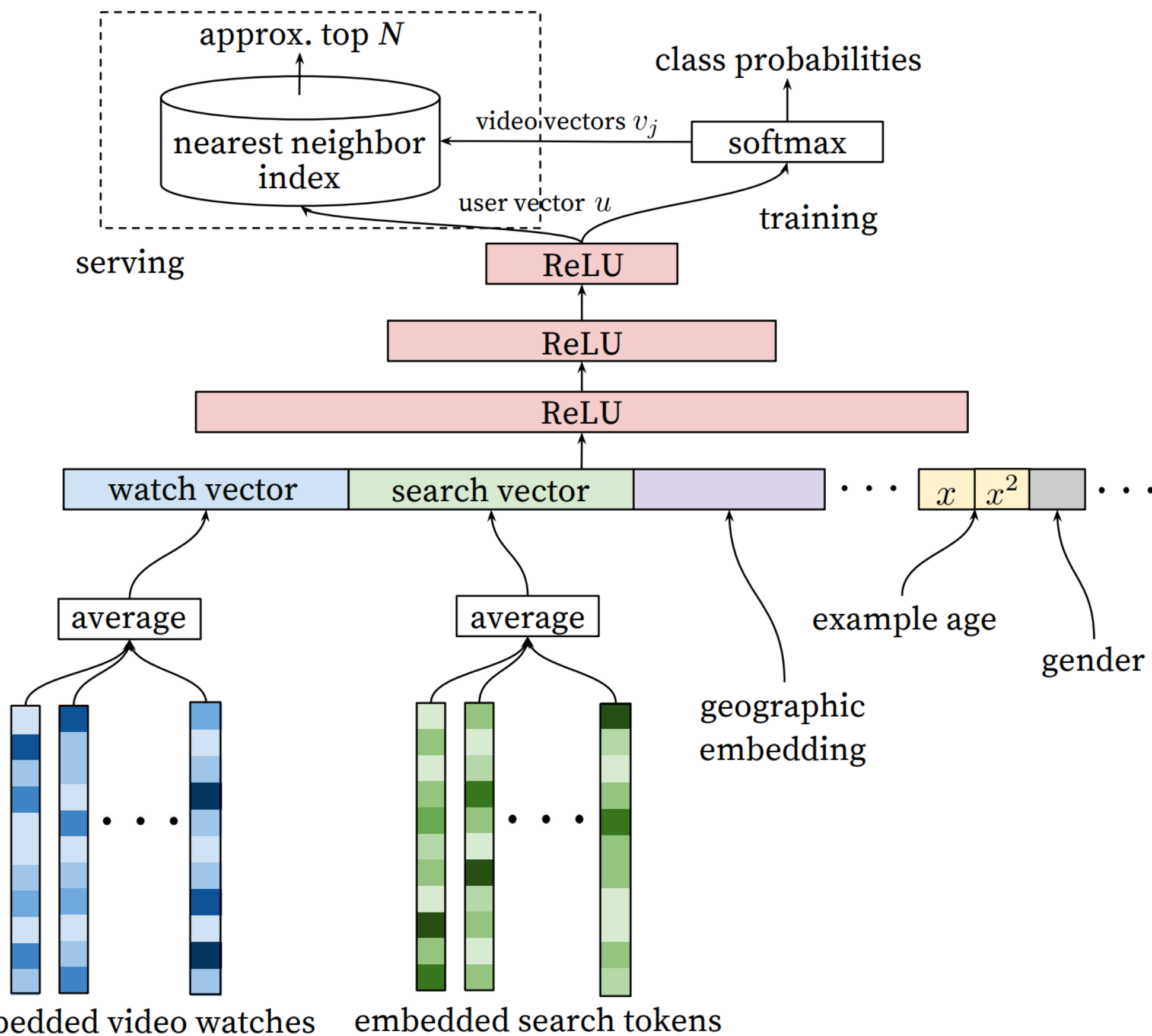
NOT Deep Structured Semantic Model



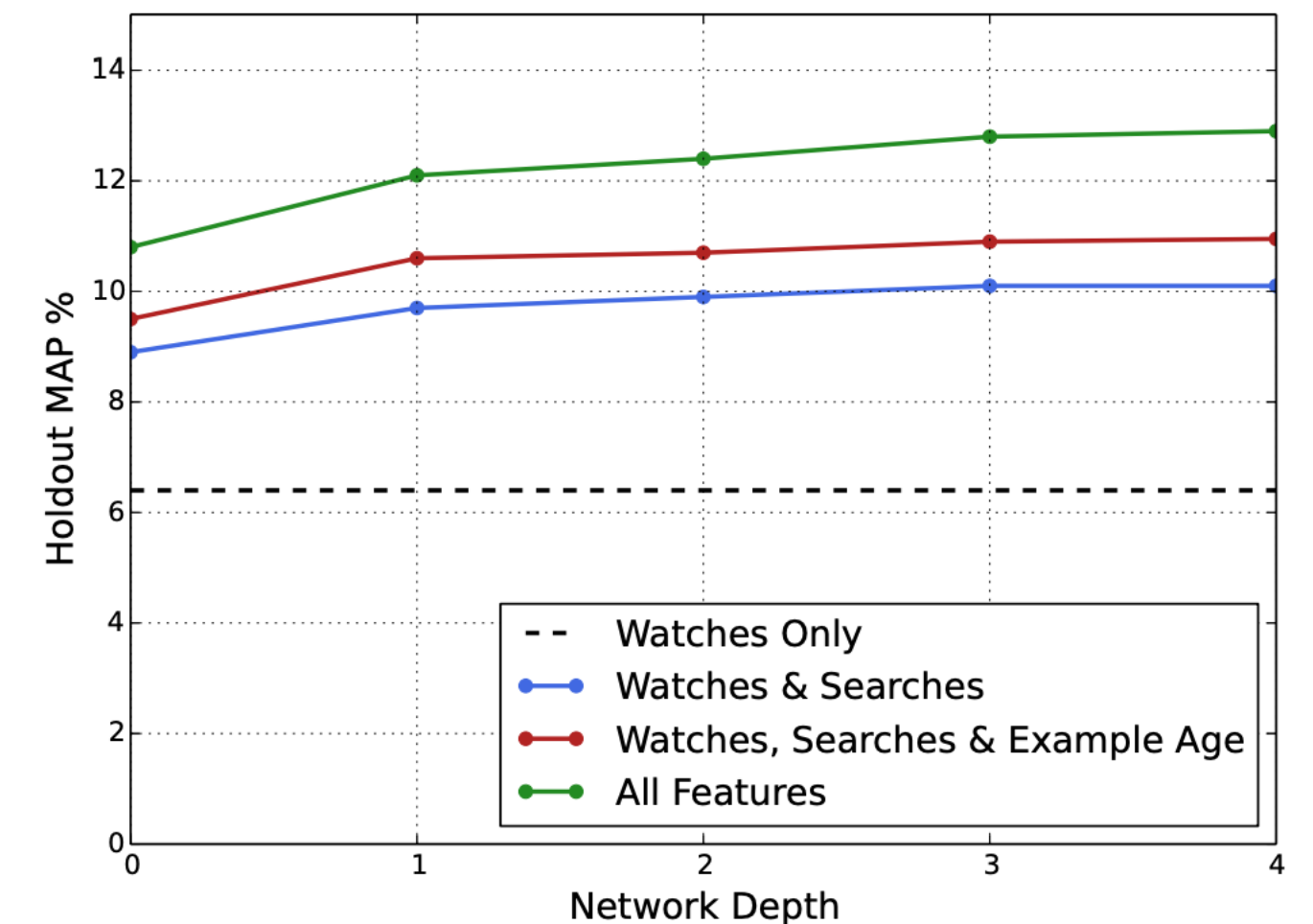
ЭТО НЕ DSSM!
ЭТО NCF



Deep Neural Networks for YouTube Recommendations



- Depth 0: A linear layer simply transforms the concatenation layer to match the softmax dimension of 256
- Depth 1: 256 ReLU
- Depth 2: 512 ReLU → 256 ReLU
- Depth 3: 1024 ReLU → 512 ReLU → 256 ReLU
- Depth 4: 2048 ReLU → 1024 ReLU → 512 ReLU → 256 ReLU



$$P(w_t = i | U, C) = \frac{e^{v_i u}}{\sum_{j \in V} e^{v_j u}}$$

Deep Neural Networks for YouTube Recommendations

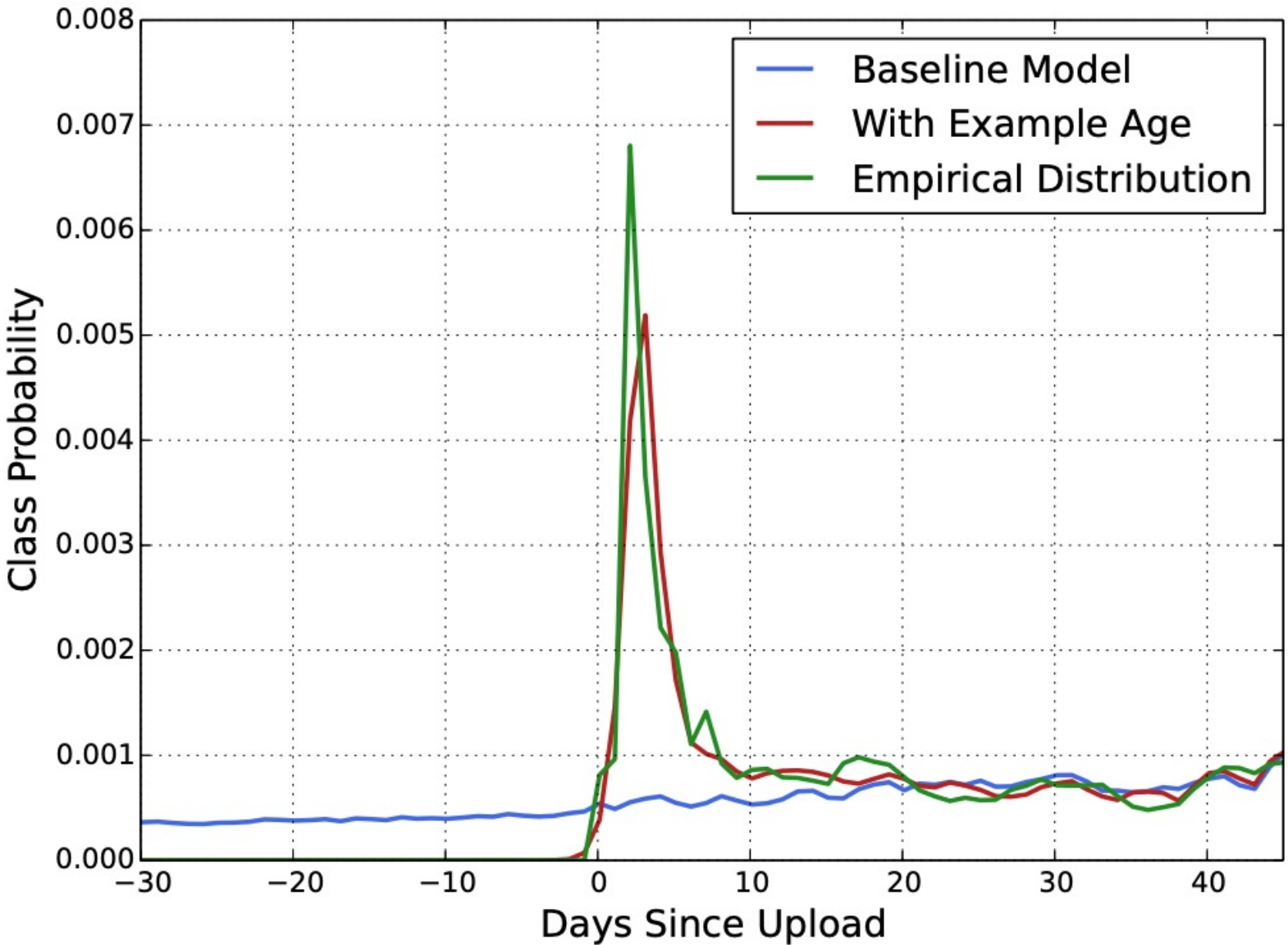


Figure 4: For a given video [26], the model trained with example age as a feature is able to accurately represent the upload time and time-dependant popularity observed in the data. Without the feature, the model would predict approximately the average likelihood over the training window.

Hidden layers	weighted, per-user loss
None	41.6%
256 ReLU	36.9%
512 ReLU	36.7%
1024 ReLU	35.8%
512 ReLU \rightarrow 256 ReLU	35.2%
1024 ReLU \rightarrow 512 ReLU	34.7%
1024 ReLU \rightarrow 512 ReLU \rightarrow 256 ReLU	34.6%

Table 1: Effects of wider and deeper hidden ReLU layers on watch time-weighted pairwise loss computed on next-day holdout data.

Deep Neural Networks for YouTube Recommendations

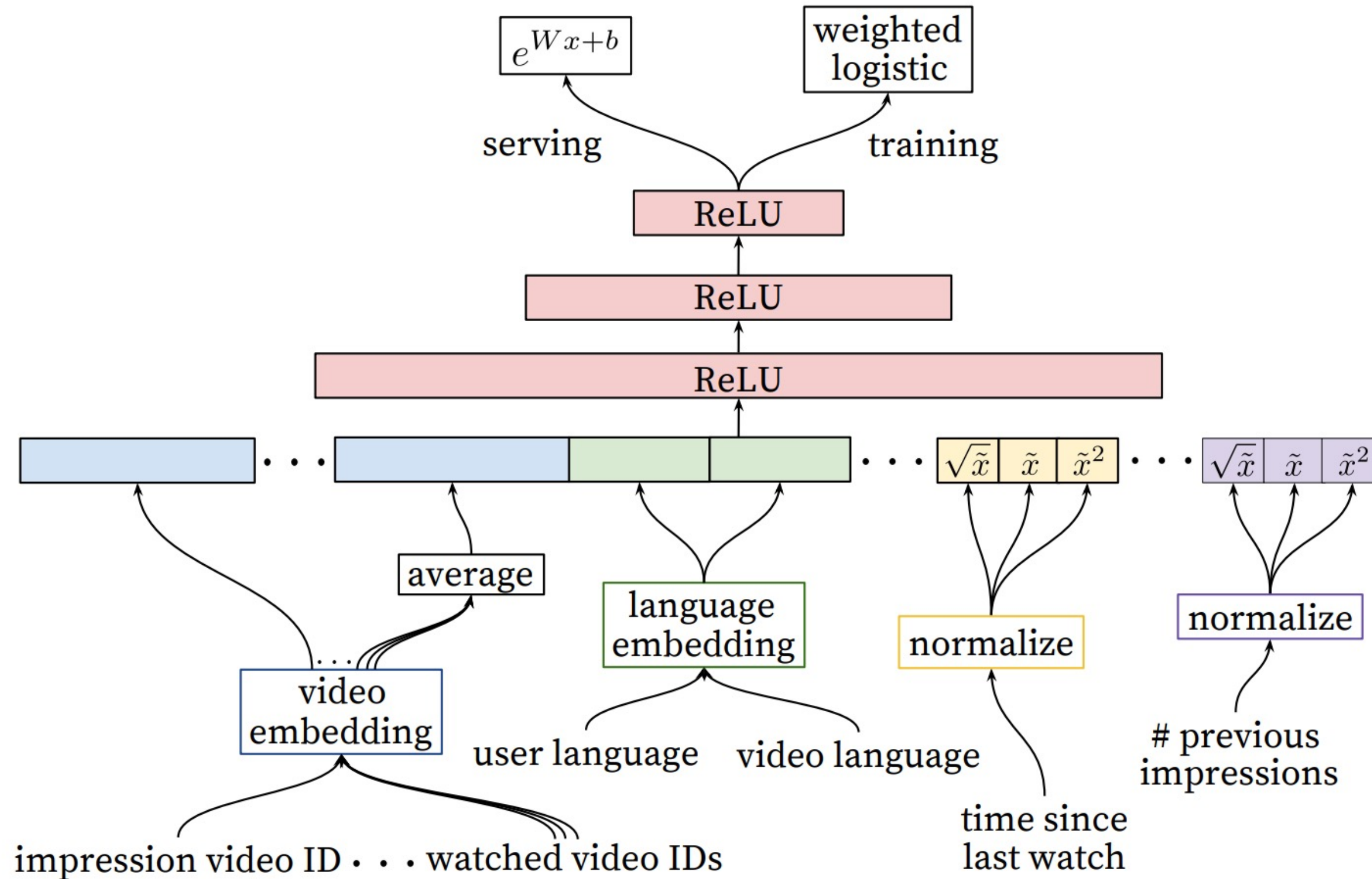


Figure 7: Deep ranking network architecture depicting embedded categorical features (both univalent and multivalent) with shared embeddings and powers of normalized continuous features. All layers are fully connected. In practice, hundreds of features are fed into the network.

Как добавить контетн/контекст в коллаборативную фильтрацию?



Как добавить контетн/контекст в коллаборативную фильрацию?



User_id	Item_id	Date
5	4	2024-02-24
6	1	2024-02-24
5	3	2024-02-22
1	2	2024-02-13
5	1	2024-02-10
4	2	2024-02-08
2	3	2024-02-08
2	1	2024-02-07
....
1	4	2023-12-31



U/I	item 1	item2	item 3	item 4
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

Как добавить контетн/контекст в коллаборативную фильрацию?

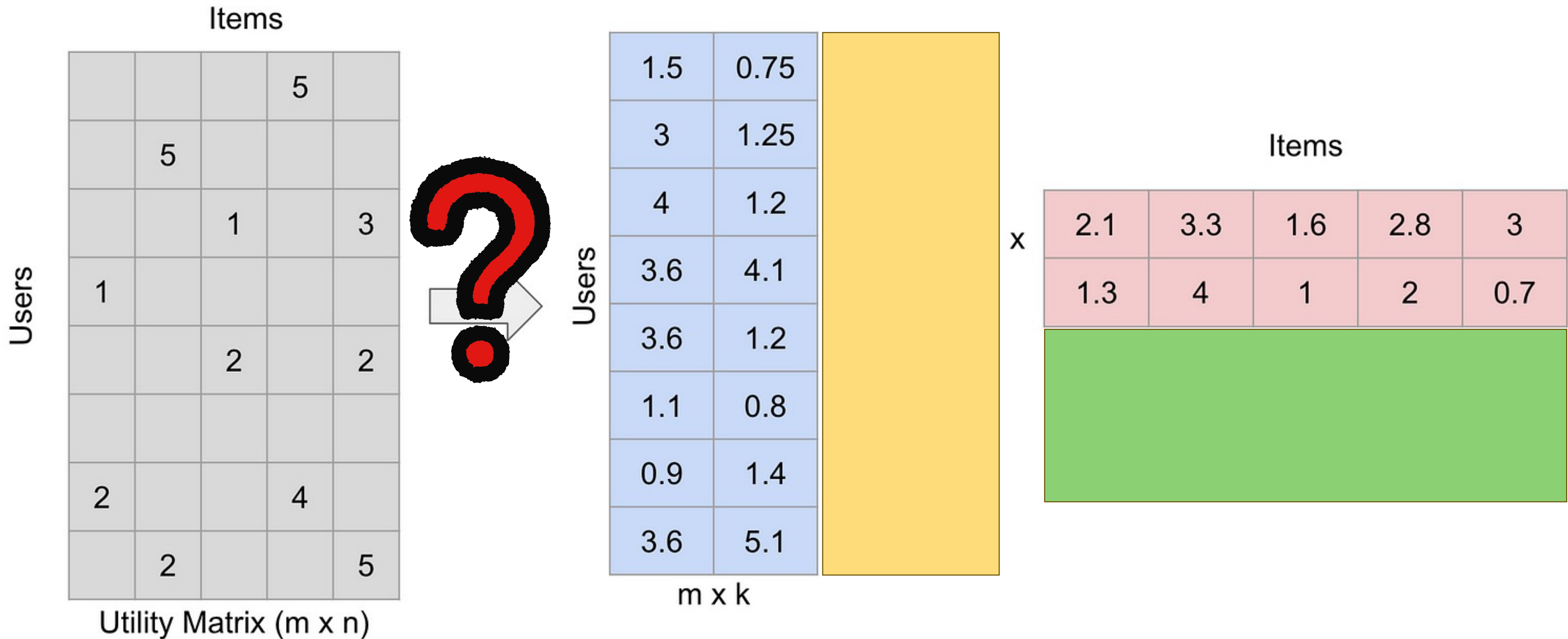


User_id	Item_id	Date
5	4	2024-02-24
6	1	2024-02-24
5	3	2024-02-22
1	2	2024-02-13
5	1	2024-02-10
4	2	2024-02-08
2	3	2024-02-08
2	1	2024-02-07
....
1	4	2023-12-31

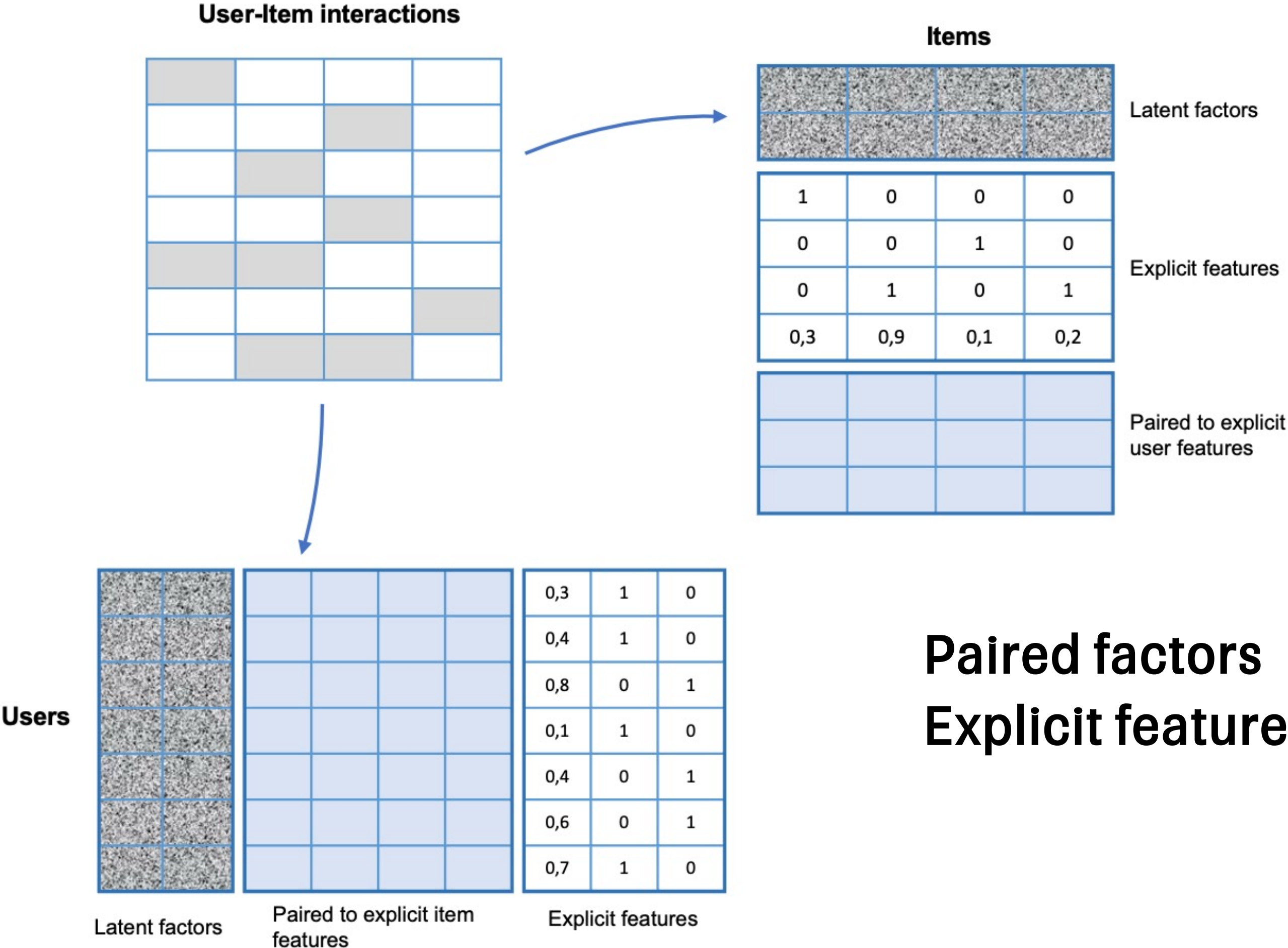


U/I	item 1	item2	item 3	item 4
User 1	0	1	0	0.5
User 2	0.5	0	1	0
User 3	0	1	0	1
User 4	0	1	0	0
User 5	0.33	0	0.5	1
User 6	1	0.33	0	0.5
User 7	0	0.5	1	0
User 8	0.5	0	0	1

Как добавить контетн/контекст в коллаборативную фильрацию?

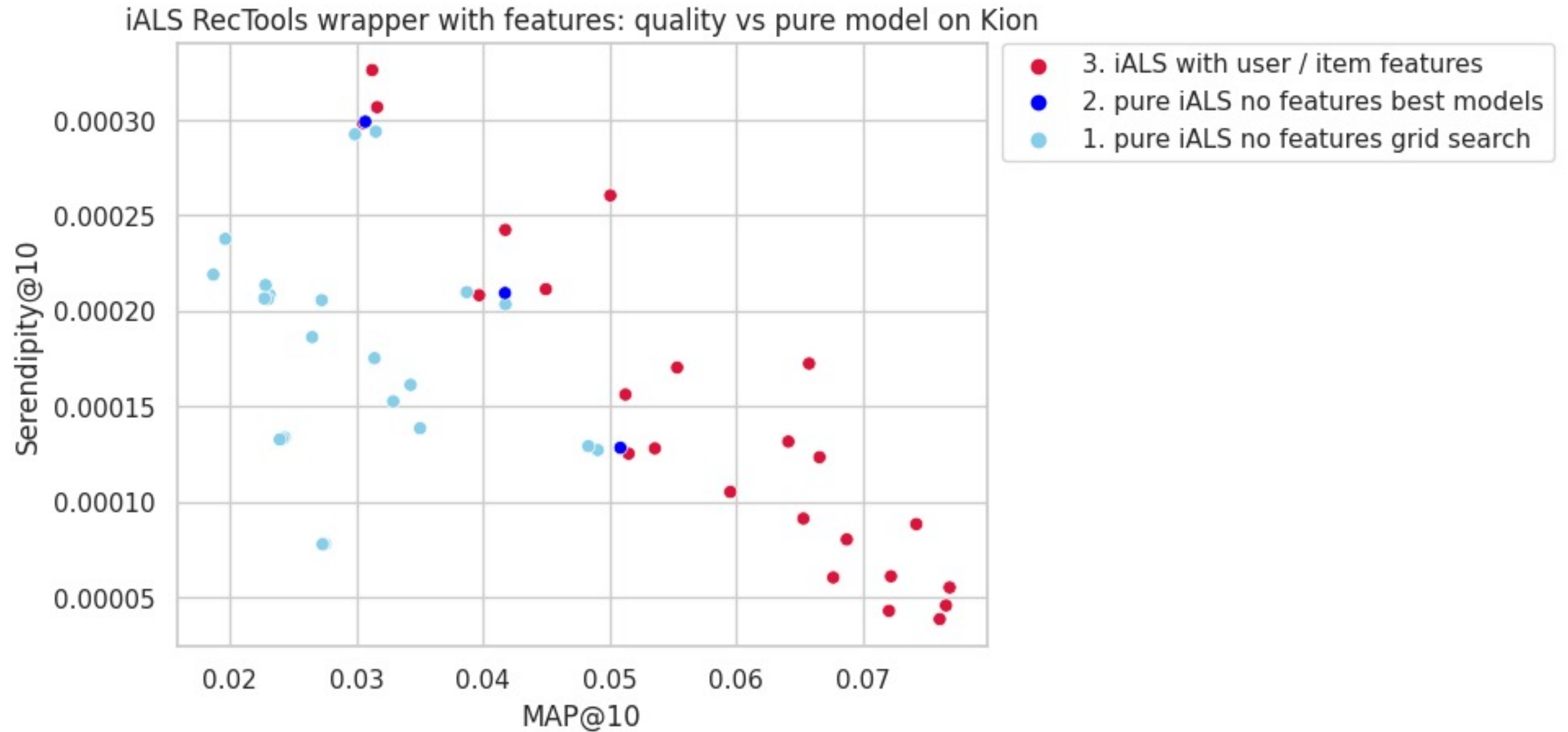


Hybrid iALS



Paired factors – обучаемые.
Explicit features – заморожены.

Content iALS



Metadata Embeddings for User and Item Cold-start Recommendations (LightFM)

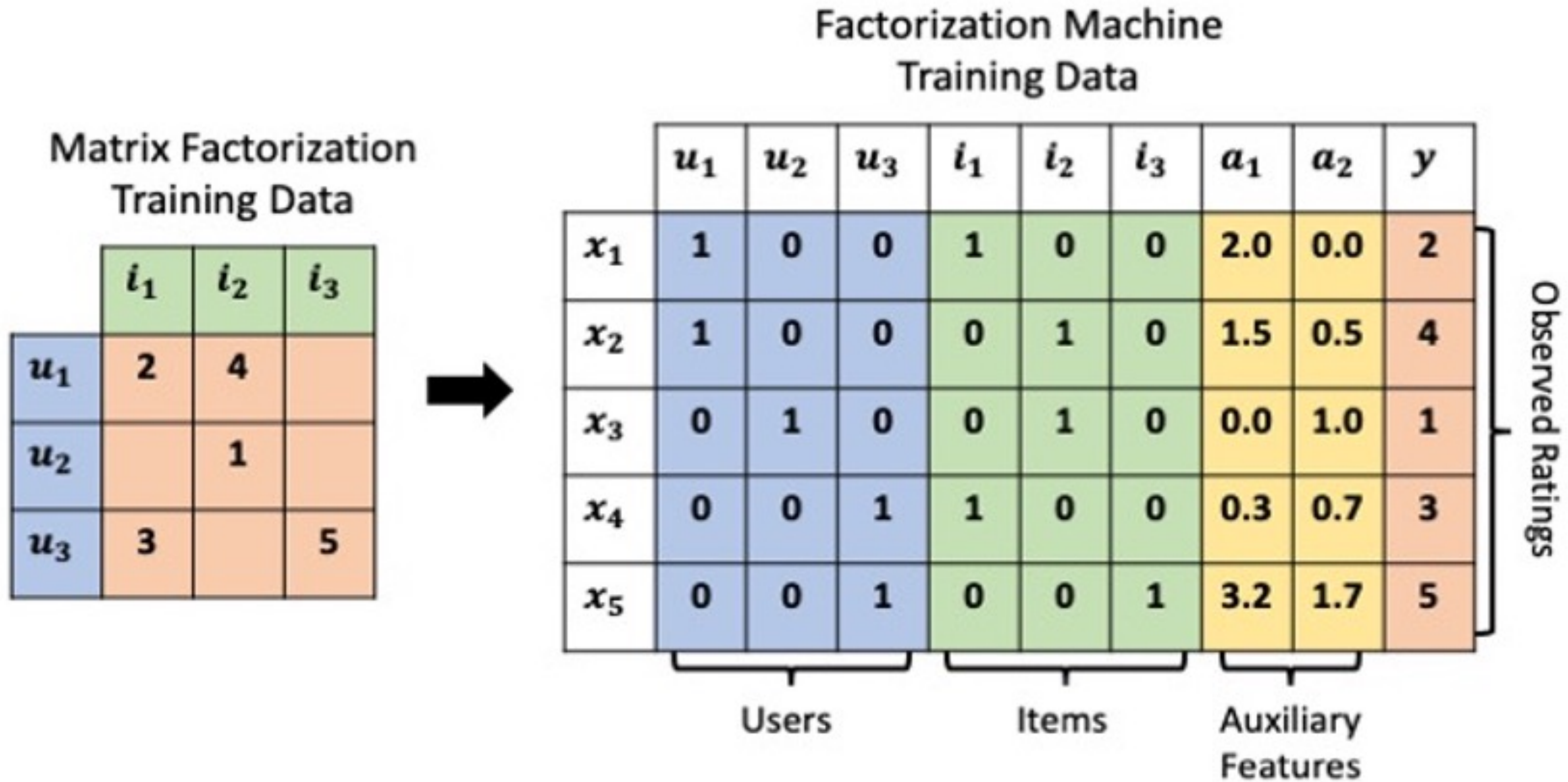


2.1 Motivation

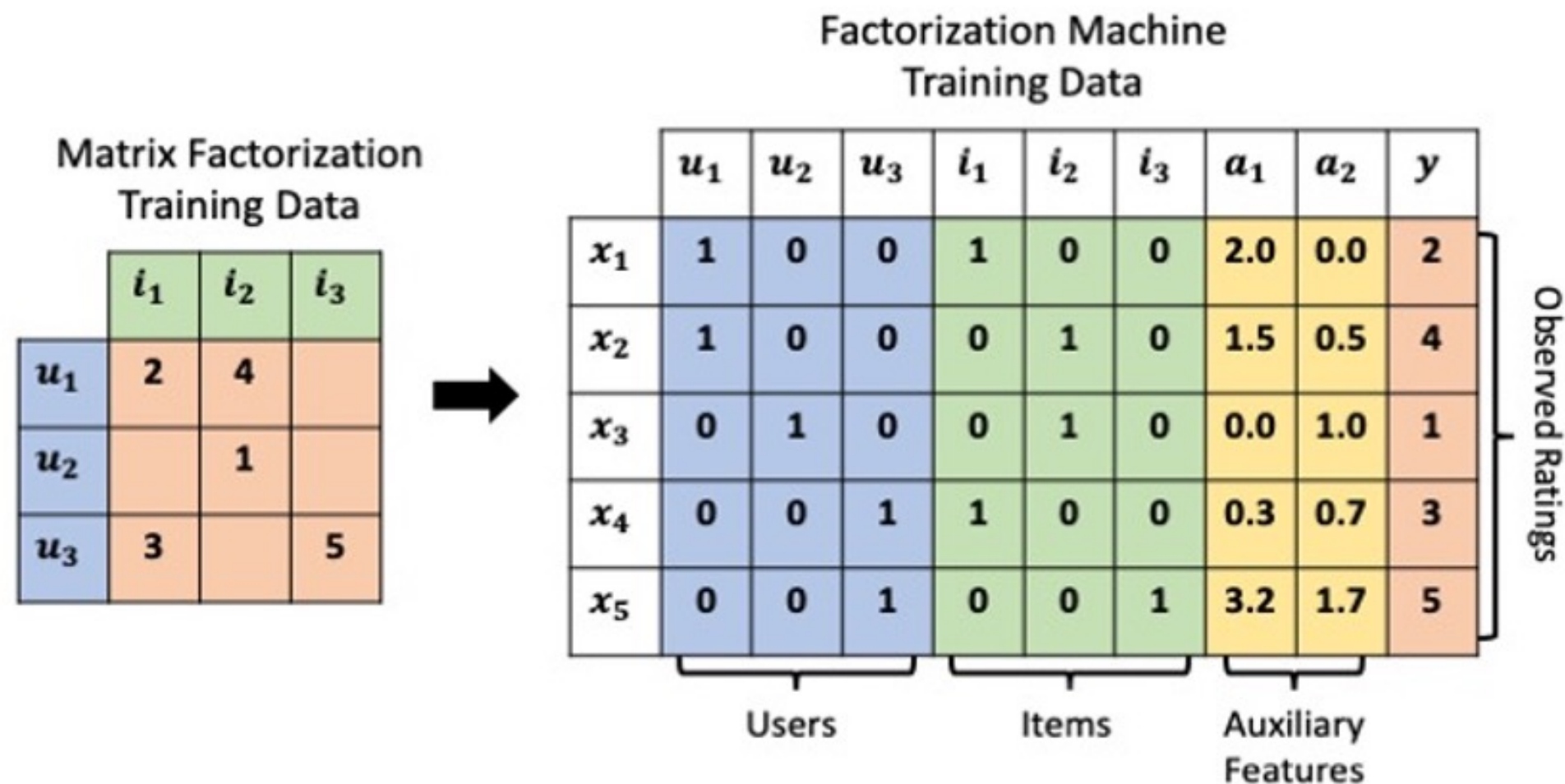
The structure of the LightFM model is motivated by two considerations.

1. The model must be able to learn user and item representations from interaction data: if items described as ‘ball gown’ and ‘pencil skirt’ are consistently all liked by users, the model must learn that ball gowns are similar to pencil skirts.
2. The model must be able to compute recommendations for new items and users.

Factorization Machines



Factorization Machines



x_p – p -я компонента вектора x_i

v_p – эмбеddинг размерности k

$$f(x) = w_0 + \sum_{p=1}^P w_p x_p + \sum_{p=1}^{P-1} \sum_{q=p+1}^P \langle v_p, v_q \rangle x_p x_q$$

Factorization Machines



Feature vector \mathbf{x}																	Target y					
$\mathbf{x}^{(1)}$	1	0	0	...	1	0	0	0	...	0.3	0.3	0.3	0	...	13	0	0	0	0	...	5	$y^{(1)}$
$\mathbf{x}^{(2)}$	1	0	0	...	0	1	0	0	...	0.3	0.3	0.3	0	...	14	1	0	0	0	...	3	$y^{(2)}$
$\mathbf{x}^{(3)}$	1	0	0	...	0	0	1	0	...	0.3	0.3	0.3	0	...	16	0	1	0	0	...	1	$y^{(2)}$
$\mathbf{x}^{(4)}$	0	1	0	...	0	0	1	0	...	0	0	0.5	0.5	...	5	0	0	0	0	...	4	$y^{(3)}$
$\mathbf{x}^{(5)}$	0	1	0	...	0	0	0	1	...	0	0	0.5	0.5	...	8	0	0	1	0	...	5	$y^{(4)}$
$\mathbf{x}^{(6)}$	0	0	1	...	1	0	0	0	...	0.5	0	0.5	0	...	9	0	0	0	0	...	1	$y^{(5)}$
$\mathbf{x}^{(7)}$	0	0	1	...	0	0	1	0	...	0.5	0	0.5	0	...	12	1	0	0	0	...	5	$y^{(6)}$
	A	B	C	...	TI	NH	SW	ST	...	TI	NH	SW	ST	...		TI	NH	SW	ST	...		
	User				Movie					Other Movies rated					Time	Last Movie rated						

Metadata Embeddings for User and Item Cold-start Recommendations (LightFM)



Формализация.

U - множество пользователей,

I - множество объектов,

F^U - множество признаков пользователей,

F^I - множество признаков объектов.

Все пары $(u, i) \in U \times I$ - это объединение всех положительных S^+ и отрицательных S^- интеракций.

Каждый пользователь описан набором заранее известных признаков (мета данных) $f_u \subset F^U$, то же самое для объектов $f_i \subset F^I$.

Metadata Embeddings for User and Item Cold-start Recommendations (LightFM)



Для каждого признака f мы задаем вектор размерности d отдельно для пользователей и отдельно для айтемов (e_f^U и e_f^I соответственно).

Латентное представление пользователя представлено суммой его латентных векторов признаков:

$$q_u = \sum_{j \in f_u} e_j^U$$

Аналогично для объектов:

$$p_i = \sum_{j \in f_i} e_j^I$$

Так же, по пользователю и объекту есть смещения (bias):

$$b_u = \sum_{j \in f_u} b_j^U$$

$$b_i = \sum_{j \in f_i} b_j^I$$

Предсказание из модели будет получать через скалярное произведение эмбедингов пользователя и объекта.

$$\hat{r}_{ui} = f(q_u \cdot p_i + b_u + b_i)$$

Metadata Embeddings for User and Item Cold-start Recommendations (LightFM)



Функция $f()$ может быть разной, автор статьи выбрал сигмоиду, поскольку использовал бинарные данные.

$$f(x) = \frac{1}{1 + \exp(-x)}$$

Задача оптимизации будет сформулирована как максимизация правдоподобия (данных при параметрах), с обучением модели с помощью стохастического градиентного спуска.

$$L(e^U, e^I, b^U, b^I) = \prod_{(u,i) \in S^+} \hat{r}_{ui} \cdot \prod_{(u,i) \in S^-} (1 - \hat{r}_{ui})$$

Стоит отметить, что оригинальная версия LightFM поддерживает только категориальные признаки (из-за механики эмбедингов)

Metadata Embeddings for User and Item Cold-start Recommendations (LightFM)



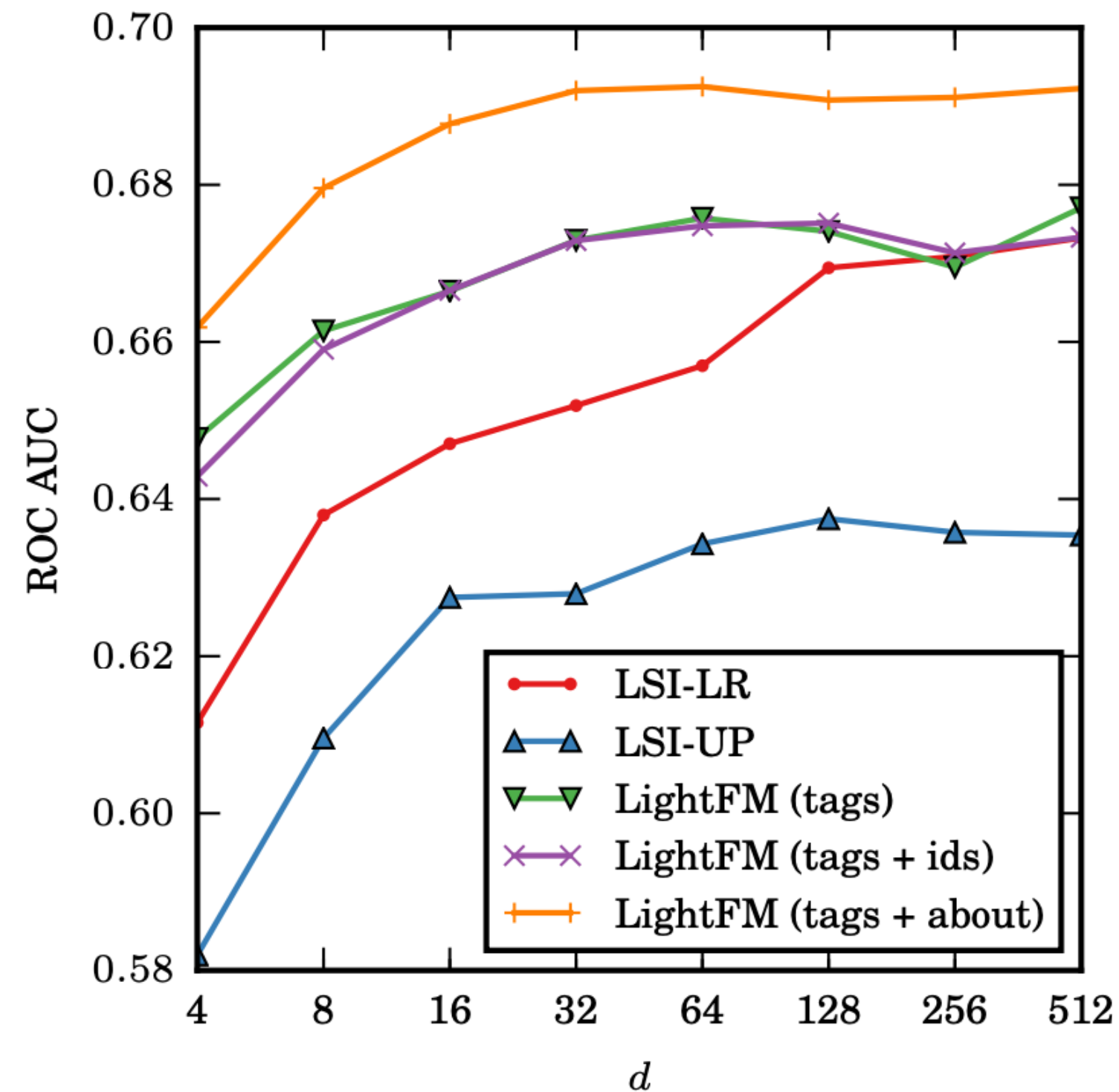
4. **LightFM (tags)**: the LightFM model using only tag features.
5. **LightFM (tags + ids)**: the LightFM model using both tag and item indicator features.
6. **LightFM (tags + about)**: the LightFM model using both item and user features. User features are available only for the CrossValidated dataset. I construct them by converting the 'About Me' sections of users' profiles to a bag-of-words representation. I first strip them of all HTML tags and non-alphabetical characters, then convert the resulting string to lowercase and tokenise on spaces.

Metadata Embeddings for User and Item Cold-start Recommendations (LightFM)

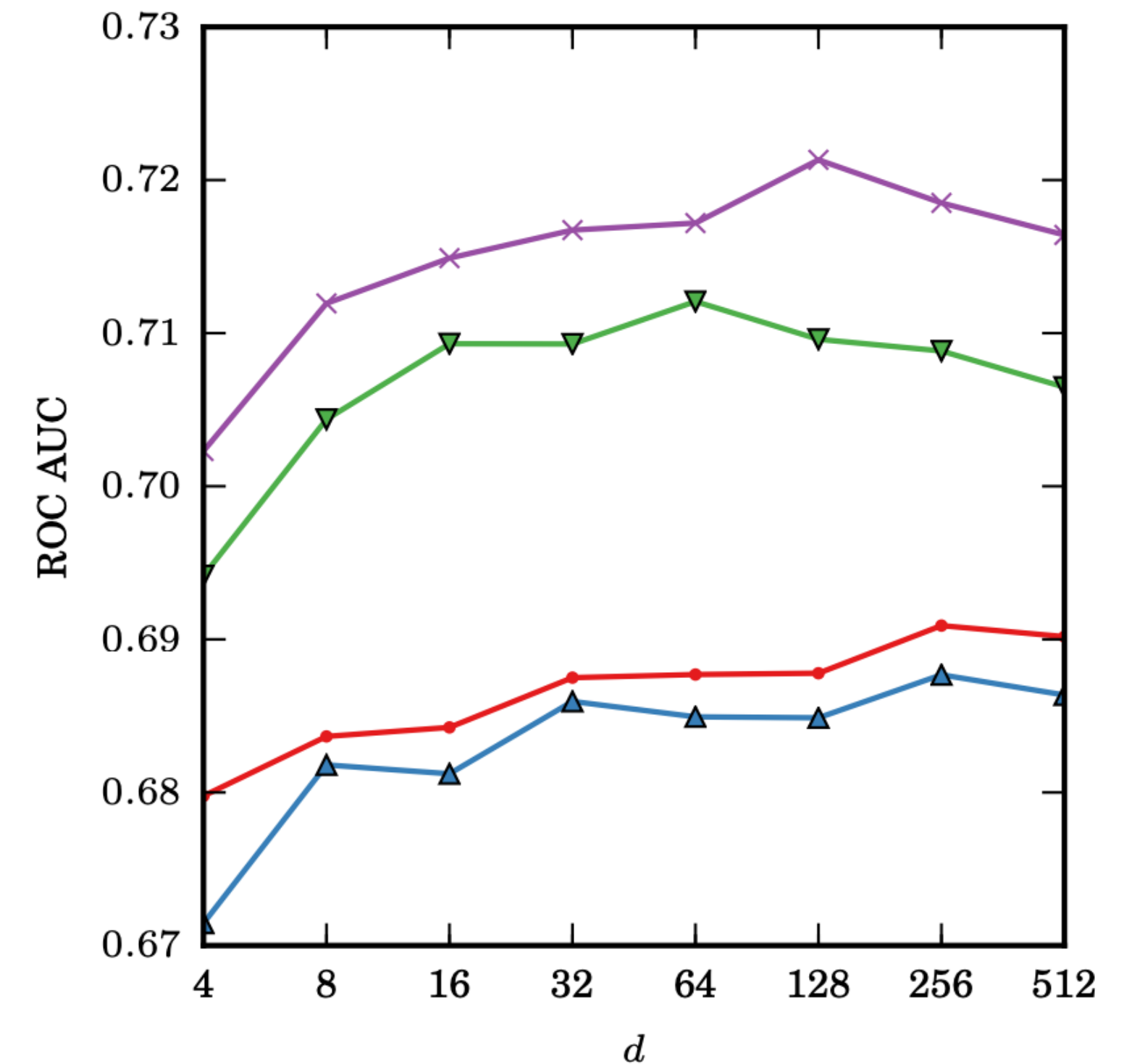


	CrossValidated		MovieLens	
	Warm	Cold	Warm	Cold
LSI-LR	0.662	0.660	0.686	0.690
LSI-UP	0.636	0.637	0.687	0.681
MF	0.541	0.508	0.762	0.500
LightFM (tags)	0.675	0.675	0.744	0.707
LightFM (tags + ids)	0.682	0.674	0.763	0.716
LightFM (tags + about)	0.695	0.696		

Metadata Embeddings for User and Item Cold-start Recommendations (LightFM)



(a) CrossValidated



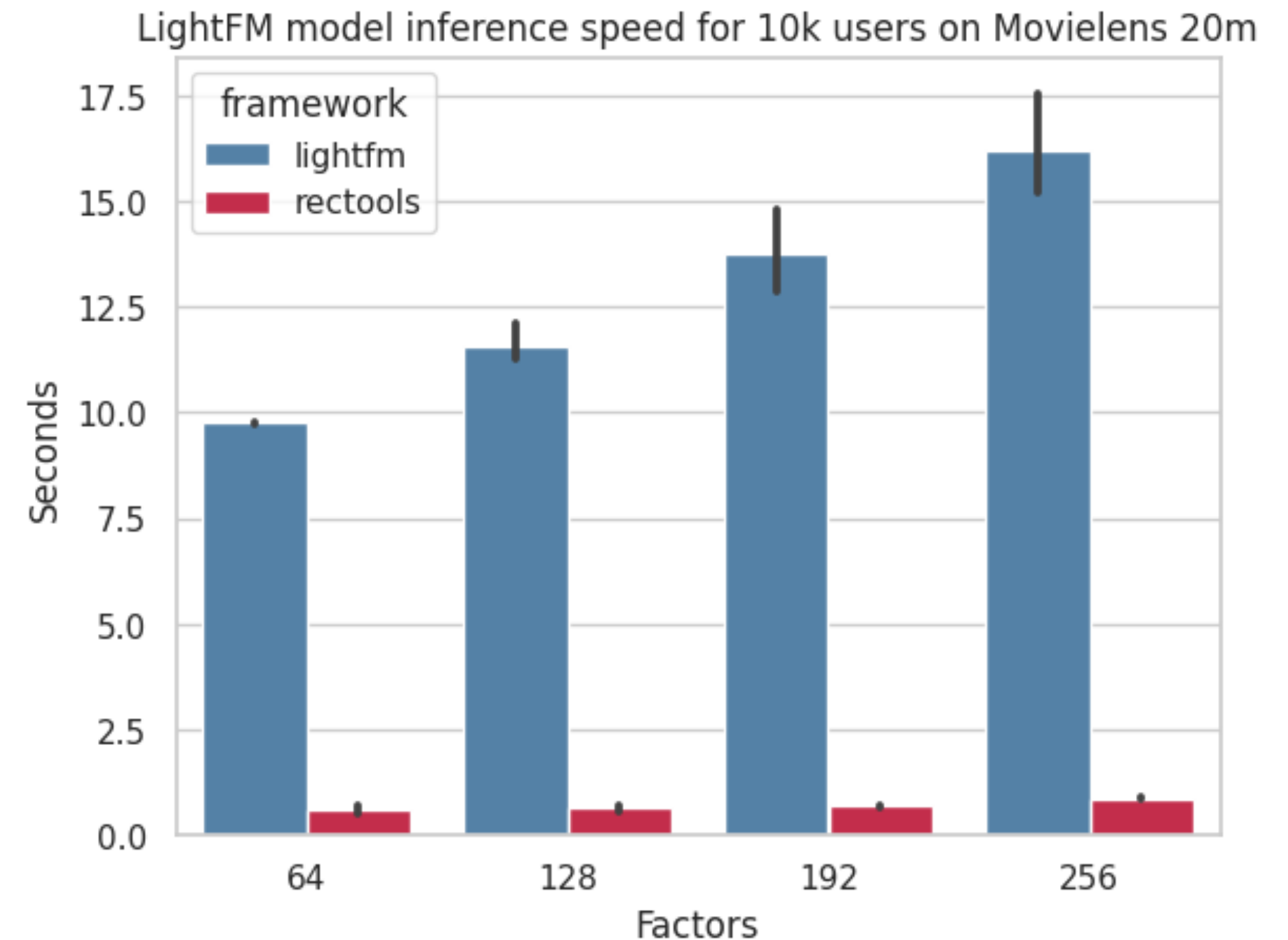
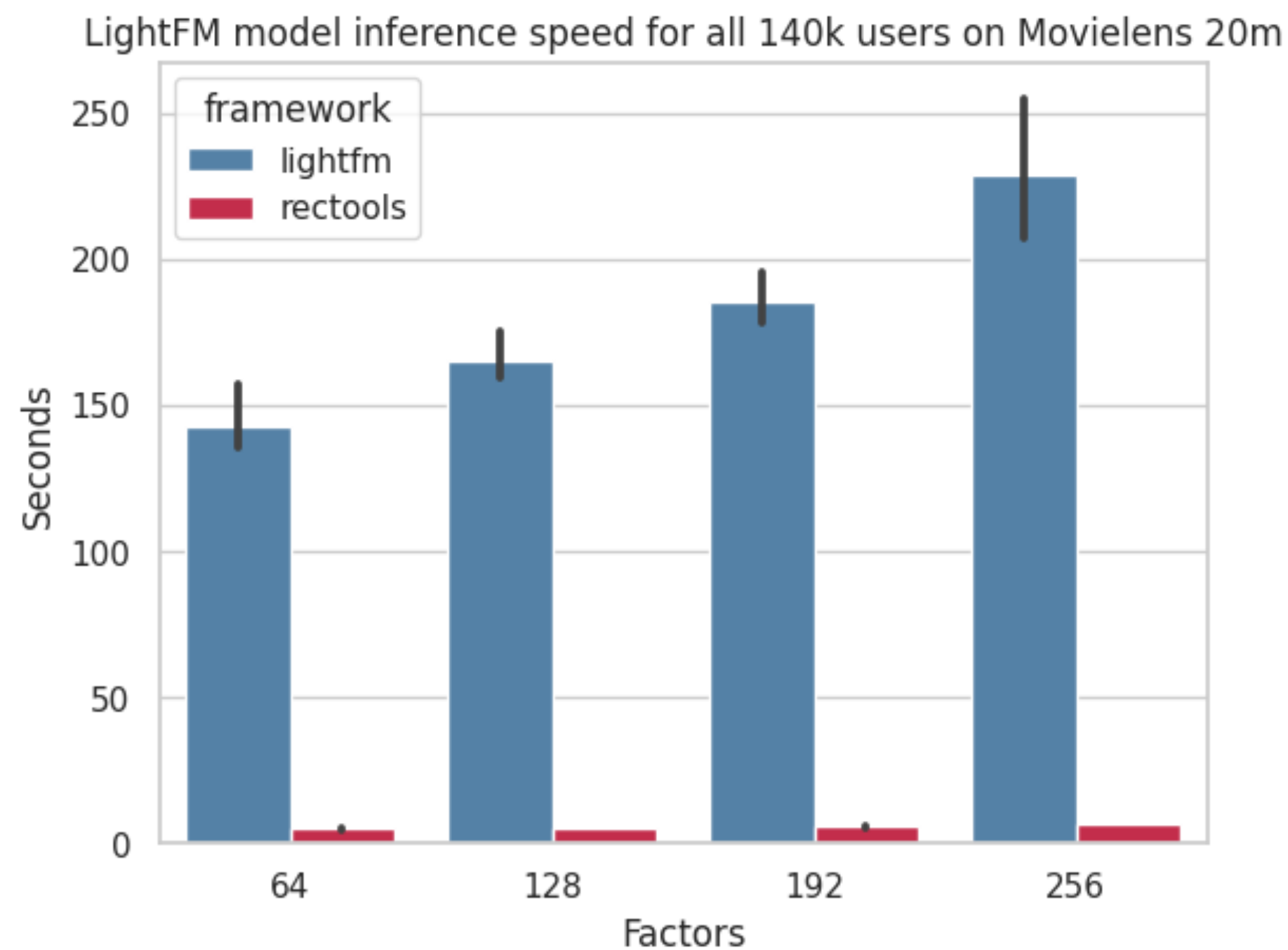
(b) MovieLens

Реализации LightFM



Оригинальная от автора статьи: <https://making.lyst.com/lightfm/docs/home.html>

Современная от МТС в библиотеке RecTools: <https://github.com/MobileTeleSystems/RecTools>



Спасибо за
внимание