

Collaborative Filtering

Турал Гурбанов (ivi.ru)

About me



About me

- Создатель рекомендательной системы ivi.ru



About me

- Создатель рекомендательной системы ivi.ru
- PhD in Recommender Systems



About me

- Создатель рекомендательной системы ivi.ru
- PhD in Recommender Systems
- Head of ML&AI в ivi.ru



About me

- Создатель рекомендательной системы ivi.ru
- PhD in Recommender Systems
- Head of ML&AI в ivi.ru
- Предпочитаю темную сторону ML



About me

- Создатель рекомендательной системы ivi.ru
- PhD in Recommender Systems
- Head of ML&AI в ivi.ru
- Предпочитаю темную сторону ML



@tgurbanov



Roadmap

Roadmap

- Intro to Recommender Systems (RS)

Roadmap

- Intro to Recommender Systems (RS)
- Explicit and implicit feedback

Roadmap

- Intro to Recommender Systems (RS)
- Explicit and implicit feedback
- Collaborative Filtering (CF)
 - Memory-based techniques
 - Model-based techniques

Roadmap

- Intro to Recommender Systems (RS)
- Explicit and implicit feedback
- Collaborative Filtering (CF)
 - Memory-based techniques
 - Model-based techniques
- Improvements and extensions

Roadmap

- Intro to Recommender Systems (RS)
- Explicit and implicit feedback
- Collaborative Filtering (CF)
 - Memory-based techniques
 - Model-based techniques
- Improvements and extensions
- Wrap-up

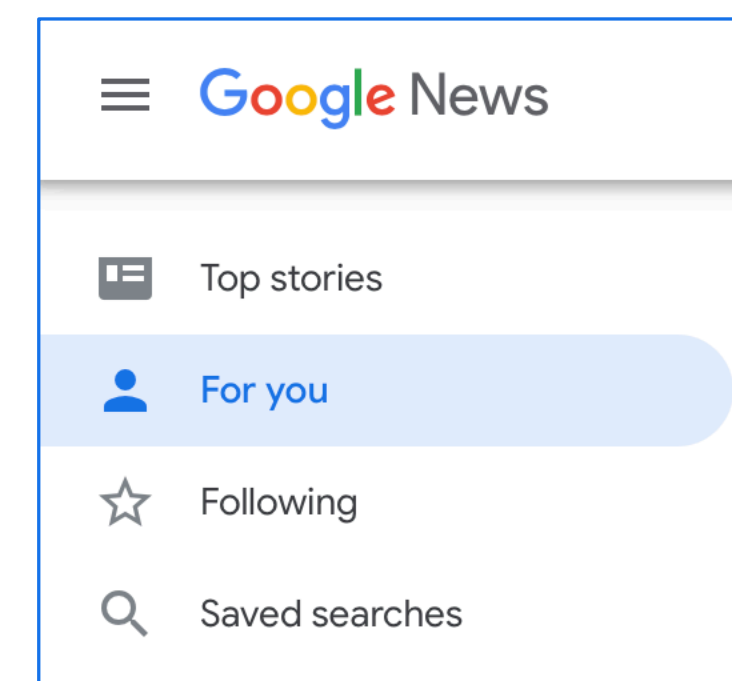
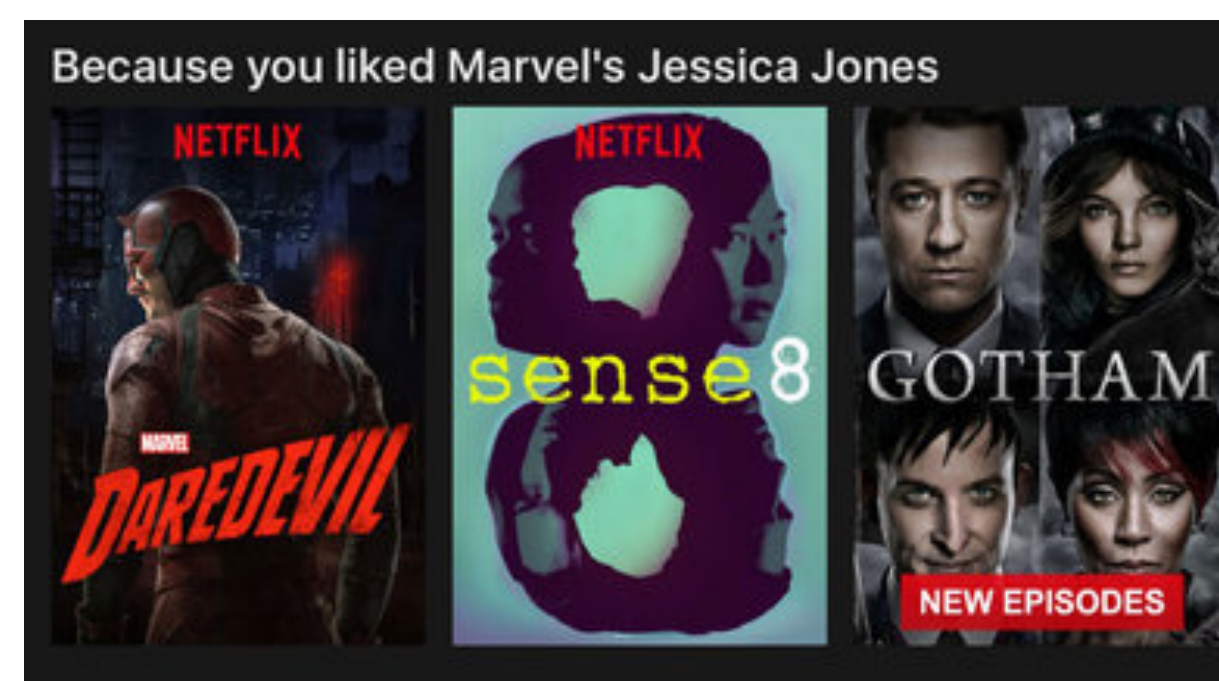
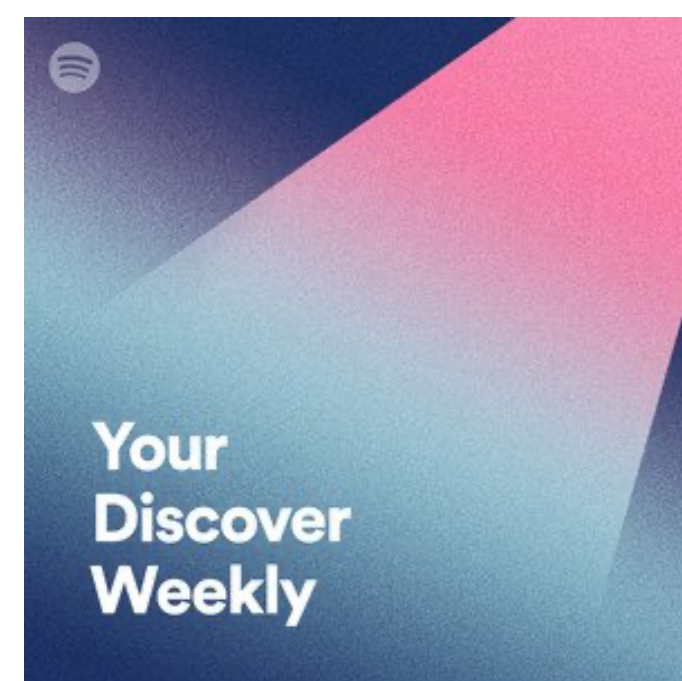
Roadmap

- Intro to Recommender Systems (RS)
- Explicit and implicit feedback
- Collaborative Filtering (CF)
 - Memory-based techniques
 - Model-based techniques
- Improvements and extensions
- Wrap-up
- Case study: ivi.ru

Introduction

Recommender Systems (RSs)

Software tools and techniques providing personalized suggestions for items that are predicted to be of interest to the user



Light & Dark Side

Light & Dark Side

- Академический — помочь пользователю найти объекты, которые будут ему интересны и которые он не смог бы найти без помощи RS

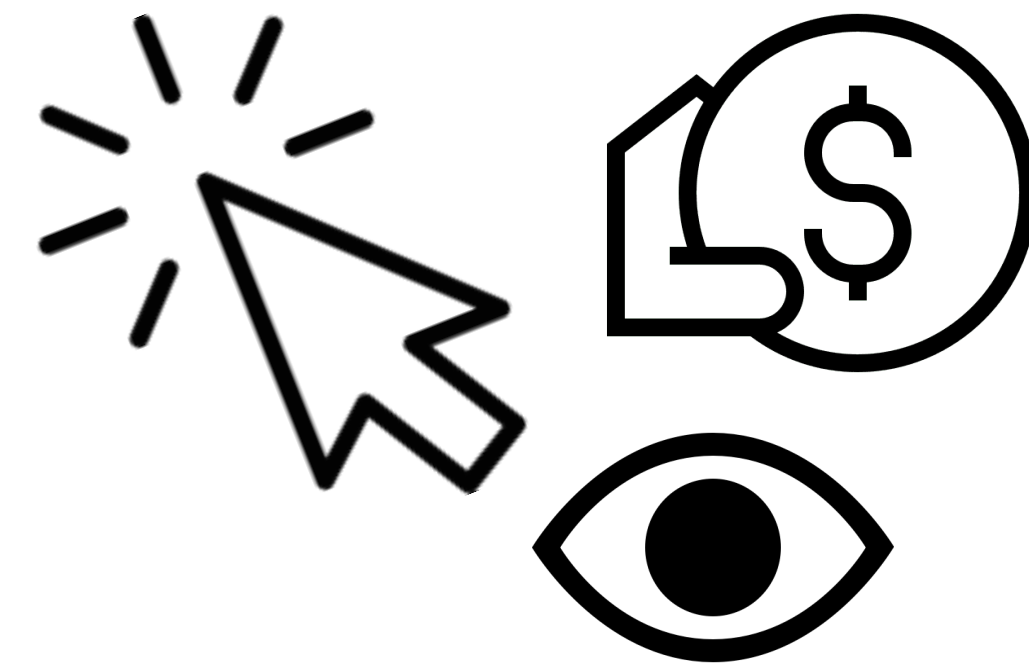
Light & Dark Side

- Академический — помочь пользователю найти объекты, которые будут ему интересны и которые он не смог бы найти без помощи RS
- Бизнесовый — показать пользователю такие объекты, которые будут ему интересны и которые оптимизируют некоторую бизнес метрику

User Preferences

RSs derive user preferences from feedback provided by users on items:

- Explicit feedback - a user provides her preferences in the form of ratings or explicit likes/dislikes
- Implicit feedback - the system infers the user's opinion from the user's action (clicks, purchases, views, etc.)



Explicit Feedback

Преимущества

- Коррелирует с пользовательским предпочтением

Недостатки

- Актуален лишь в момент отзыва
- Требуется дополнительных действий от пользователя
- Явных отзывов обычно мало
- Доступен лишь в некоторых областях

Implicit Feedback

Преимущества

- Available in many different domains
- No need for users to invest time in expressing their preferences
- Every user interaction with the system is a potential feedback

Недостатки

- Only indirectly signals users' preferences and opinions
- The interpretation of the same type of interactions may vary

Collaborative Filtering

Исходная задача

- Есть информация о том, какие объекты нравились пользователям в прошлом
- Хотим создать некоторую функцию F , которая предсказывает для неизвестной пары user-item “уровень предпочтения”:

$$F: \text{User} \times \text{Item} \rightarrow \text{Utility}$$

Content-based RS (CBRS)

- Зависит от качества описания объектов
- Чрезмерная специализация
 - Рекомендации для пользователя основаны на знании о предпочтениях только этого пользователя
- В базовой комплектации CBRS не может рекомендовать сопутствующие товары

Чего мы хотим?

Сохранить преимущества CBRS

- Пользователь предпочитает объекты, похожие на те, которые ему понравились ранее

Учесть недостатки CBRS

- Чрезмерная специализация из-за привязывания к свойствам объектов

Добавить нового

- Пользователи с похожим вкусом предпочитают похожие объекты

То есть...

Мы хотим модели, которые строят персональные рекомендации на основе паттернов рейтингов или действий (например, просмотров), без необходимости использования дополнительной информации о свойствах товарах или пользователях.

Добро пожаловать в Collaborative Filtering 🎉!

Basic techniques

- Memory-based
 - User-based CF
 - Item-based CF
- Model based
 - Matrix Factorization (MF)
 - Weighted-Regularized MF (WR-MF)

Matrix completion

Observation matrix R

	i_1	i_2	i_3	i_4	
u_1	?	2	3	?	$\begin{array}{c} \uparrow \\ \text{user} \\ \downarrow \end{array}$
u_2	1	?	?	2	
u_3	2	1	?	?	
u_4	?	?	4	3	
u_5	?	?	1	?	
	\longleftrightarrow item \longrightarrow				

R is a user-item matrix of size $|U| \times |I|$

- U – is a set of users
- I – is a set of items

r_{ui} – indicates the rating value given by u to i
(e.g., u_3 scored i_1 by two stars)

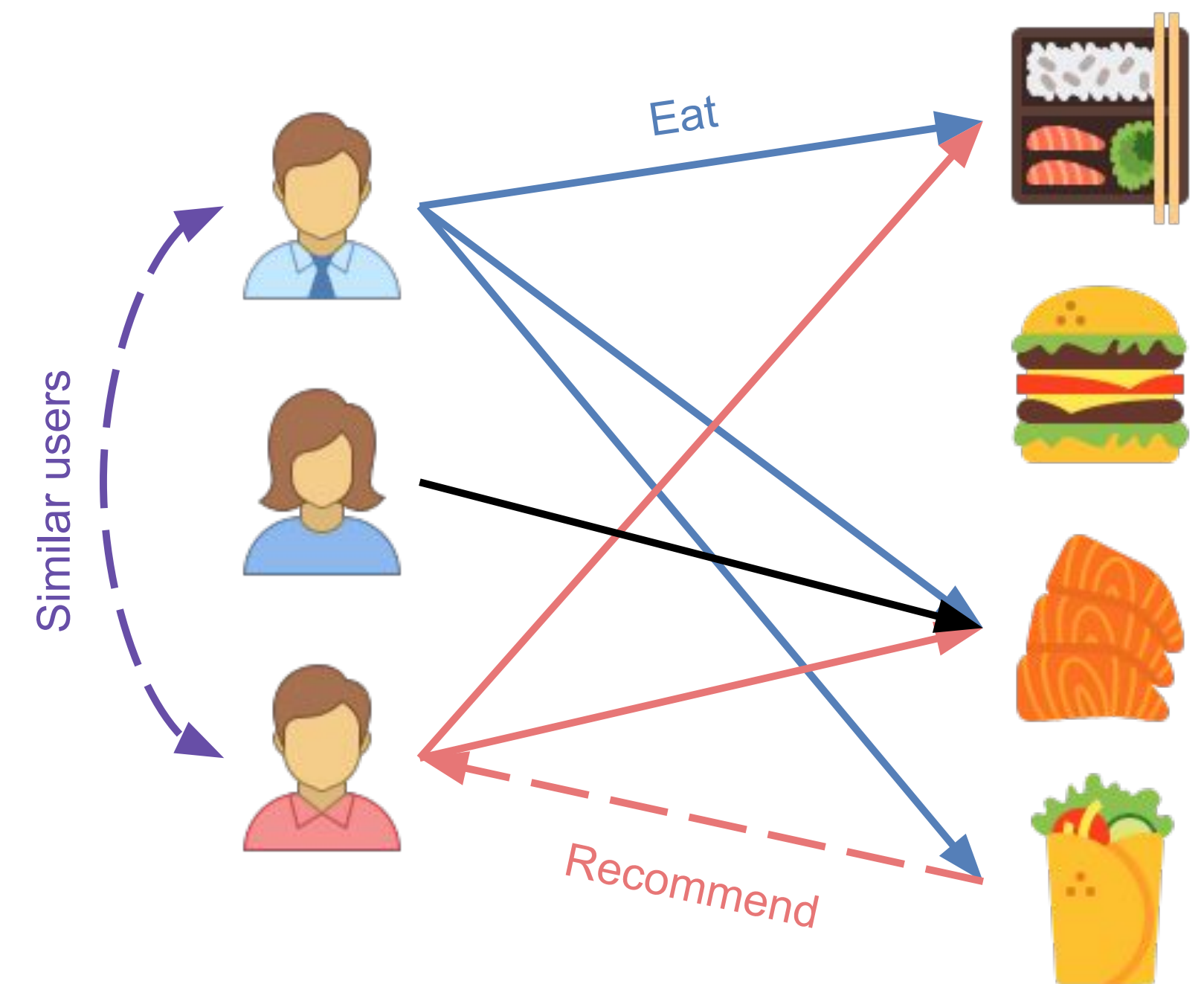
The task is to fill-in the missing entries (i.e., to predict how u_4 will rate i_2)

Memory-based Techniques

The observations stored in the system are directly used to predict unknown observations.

User-based CF

- Evaluate the interest of a target user for an item using the user-item interactions for this item generated by other users, called neighbors, that have similar interaction patterns.
- The neighbors of the target user are typically the users whose interactions are mostly correlated to the target user's interactions.



User-based CF

Базовый

$$\hat{r}_{ui} = \frac{\sum_{v \in N(u)} sim_{uv} * r_{vi}}{\sum_{v \in N(u)} sim_{uv}}$$

Нормализованный

$$\hat{r}_{ui} = \bar{r}_u + \sigma_u * \frac{\sum_{v \in N(u)} sim_{uv} * \frac{r_{vi} - \bar{r}_v}{\sigma_v}}{\sum_{v \in N(u)} sim_{uv}}$$

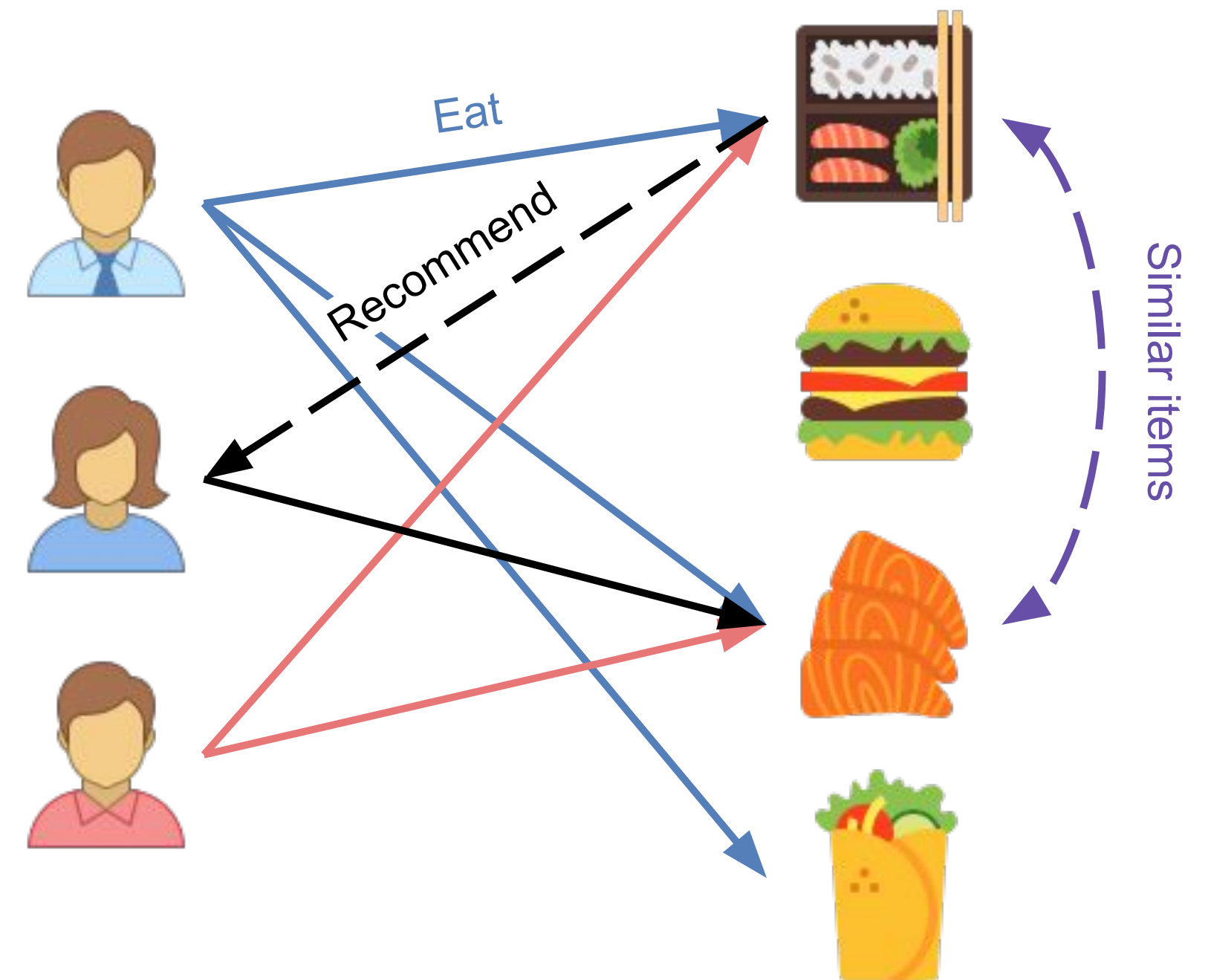
Общий

$$\hat{r}_{ui} = h^{-1} \left(\frac{\sum_{v \in N(u)} sim_{uv} * h(r_{vi})}{\sum_{v \in N(u)} sim_{uv}} \right)$$

Item-based CF

- Predict the interaction of a user with an item based on the interactions of the user with similar items.
- In such approaches, two items are similar if several users of the system have interacted with these items in a similar fashion.

$$\hat{r}_{ui} = h^{-1}\left(\frac{\sum_{j \in N(i)} sim_{ij} * h(r_{uj})}{\sum_{j \in N(i)} sim_{ij}}\right)$$



Как найти соседей?

- k-Nearest Neighbors (kNN)
 - Выбор k значительно влияет на качество рекомендаций
 - Оптимальное значение k зависит от данных (обычно $k \in [25, 50]$)
 - Функция близости: cosine similarity, коэффициент корреляции Пирсона и т.д.
- Approximate kNN
 - Annoy — <https://github.com/spotify/annoy>
 - faiss — <https://github.com/facebookresearch/faiss>

User-based vs. Item-based

	User-based	Item-based
Accuracy	Точнее, если пользователей меньше, чем объектов	Точнее, если пользователей больше, чем объектов
Efficiency	Эффективнее, если пользователей меньше, чем объектов	Эффективнее, если пользователей больше, чем объектов
Stability	Стабильнее, если множество пользователей изменяется редко	Стабильнее, если множество объектов изменяется редко
Justifiability	Рекомендации не очень легко объяснить пользователю	Рекомендации легко объяснить пользователю
Serendipity	Случайности вероятны	Случайности маловероятны

Объяснение рекомендаций

User-based

“Объект нравится пользователям [X и Y], чьи вкусы совпадают с вашими”

Item-based

“Объект похож на понравившиеся вам ранее объекты X и Y”

Pros & Cons

Преимущества

- Простая реализация
- Рекомендации возможно объяснить
- Редко требует большого количества ресурсов
- Для работы с новыми оценками не требуется перестраивать модель

Недостатки

- Частичное покрытие каталога
- Уязвимы к cold-start и data sparsity problems
- Для моделирования сложных зависимостей между пользователями и объектами требуются дополнительные извращения

Model-based Techniques

The observations stored in the system are used to learn a set of model parameters (e.g., latent features of the users and the items)

Matrix completion

Observation matrix R

	i_1	i_2	i_3	i_4	
u_1	?	2	3	?	<div style="display: flex; align-items: center; justify-content: center;"> <div style="margin-right: 5px;">↑</div> <div style="writing-mode: vertical-rl; transform: rotate(180deg);">user</div> <div style="margin-left: 5px;">↓</div> </div>
u_2	1	?	?	2	
u_3	2	1	?	?	
u_4	?	?	4	3	
u_5	?	?	1	?	
	<div style="display: flex; align-items: center; justify-content: center;"> <div style="margin-right: 5px;">←</div> <div>item</div> <div style="margin-left: 5px;">→</div> </div>				

R is a user-item matrix of size $|U| \times |I|$

- U – is a set of users
- I – is a set of items

r_{ui} – indicates the rating value given by u to I

Предположим, что, на основе известных нам данных, мы можем найти такие вектора x_u и y_i , что:

$$\hat{r}_{ui} = x_u^T y_i, \quad x_u \in \mathbb{R}^f \text{ and } y_i \in \mathbb{R}^f$$

Matrix Factorization (MF)

$$\hat{r}_{ui} = x_u^T y_i, \quad x_u \in \mathbb{R}^f \text{ and } y_i \in \mathbb{R}^f$$

$$\min_{x^*, y^*} \sum_{u,i} (r_{ui} - x_u^T y_i)^2 + \lambda \left(\sum_u \|x_u\|^2 + \sum_i \|y_i\|^2 \right)$$

- x_u, y_i - векторы пользователя и объекта в пространстве скрытых фич
- λ - параметр регуляризации (находится через cross-validation)

Stochastic Gradient Descent

- Позволяет вычислить параметры x^* и y^*
- Итеративный метод (ограничивается количеством итераций или допустимым значением ошибки)
- Изначально x^* и y^* заполняются случайными значениями
- На каждой итерации SGD выбирает случайную u, i пару из тренировочной выборки и обновляет x^* и y^*

$$x_u \leftarrow x_u + \gamma(e_{ui}y_i - \lambda x_u)$$

$$e_{ui} = r_{ui} - \hat{r}_{ui}$$

$$y_i \leftarrow y_i + \gamma(e_{ui}x_u - \lambda y_i)$$

γ - коэффициент обучения

Implicit Feedback CF

Basic Assumptions

- There are user behavioral patterns that highly correlate with explicit user feedback [Konstan et al. 1997, Morita et al. 1994, Oard et al. 2001]
- If a user performs a target action on an item, then she is interested in this item
 - A news article can be considered as a good recommendation for a user if the user read it

Typical Implicit Feedback Model

- Examines users' actions
- Predicts on which items the user will perform a target action
- Considers these items as good recommendations

Predicting Actions

Input: a set of observed user-item interactions of a single type (e.g., purchase)

Output: whether a user will perform the target action (i.e., purchase) on an item

One of the most commonly used single action type prediction models is

Weighted-Regularized Matrix Factorization (WR-MF)

[Hu et al. 2008, Pan et al. 2008]

WR-MF (implicit/ALS/implicit ALS)

Indicator function

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

Confidence function

$$c_{ui} = \begin{cases} 1 + \alpha r_{ui} & r_{ui} > 0 \\ 1 & \text{otherwise} \end{cases}$$

The target action prediction is computed using MF

$$\hat{p}_{ui} = x_u^T y_i$$

$x_u \in \mathbb{R}^f, y_i \in \mathbb{R}^f$ f -dimensional user and item latent features vectors

$$\min_{x^*, y^*} \sum_{u, i} c_{ui} (p_{ui} - x_u^T y_i)^2 + \lambda \left(\sum_u \|x_u\|^2 + \sum_i \|y_i\|^2 \right)$$

Alternating Least Squares (ALS)

- Если нам известны все значения матрицы, которую мы пытаемся аппроксимировать, процесс оптимизации можно ускорить
- Попеременно считая X и Y константами, аналитическим способом находим оптимальные значения для X и Y

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

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

- C_u - диагональная матрица, размера item x item ($C^u_{ij} = c_{ui}$)
- C_i - диагональная матрица, размера user x user ($C^i_{uu} = c_{ui}$)

Pros & Cons

Преимущества

- Промышленный стандарт :)
- Более точные чем memory-based (что не всегда хорошо, склонны к переобучению)

Недостатки

- Сложно интерпретировать
- Уязвимы к cold-start и data sparsity problems
- Линейная модель
- Требуется обучение моделей

Improvements & Extensions

Fold-in

По-умолчанию, MF модели не умеют работать с новыми пользователями/наблюдениями

- Чтобы их добавить, нужно обучать модель заново

В 2008 году Rendle et al., предложили технологию fold-in, которая:

- Позволяет создавать/обновлять вектора скрытых фич на лету
- Применима ко всем моделям на базе MF
- Идеино похожа на обновление вектора пользователя в WR-MF

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

Prediction vs. Ranking

- Prediction — минимизируем ошибку между предсказанным и ground truth значениями для объекта
- Ranking — пытаемся предсказать правильный порядок объектов

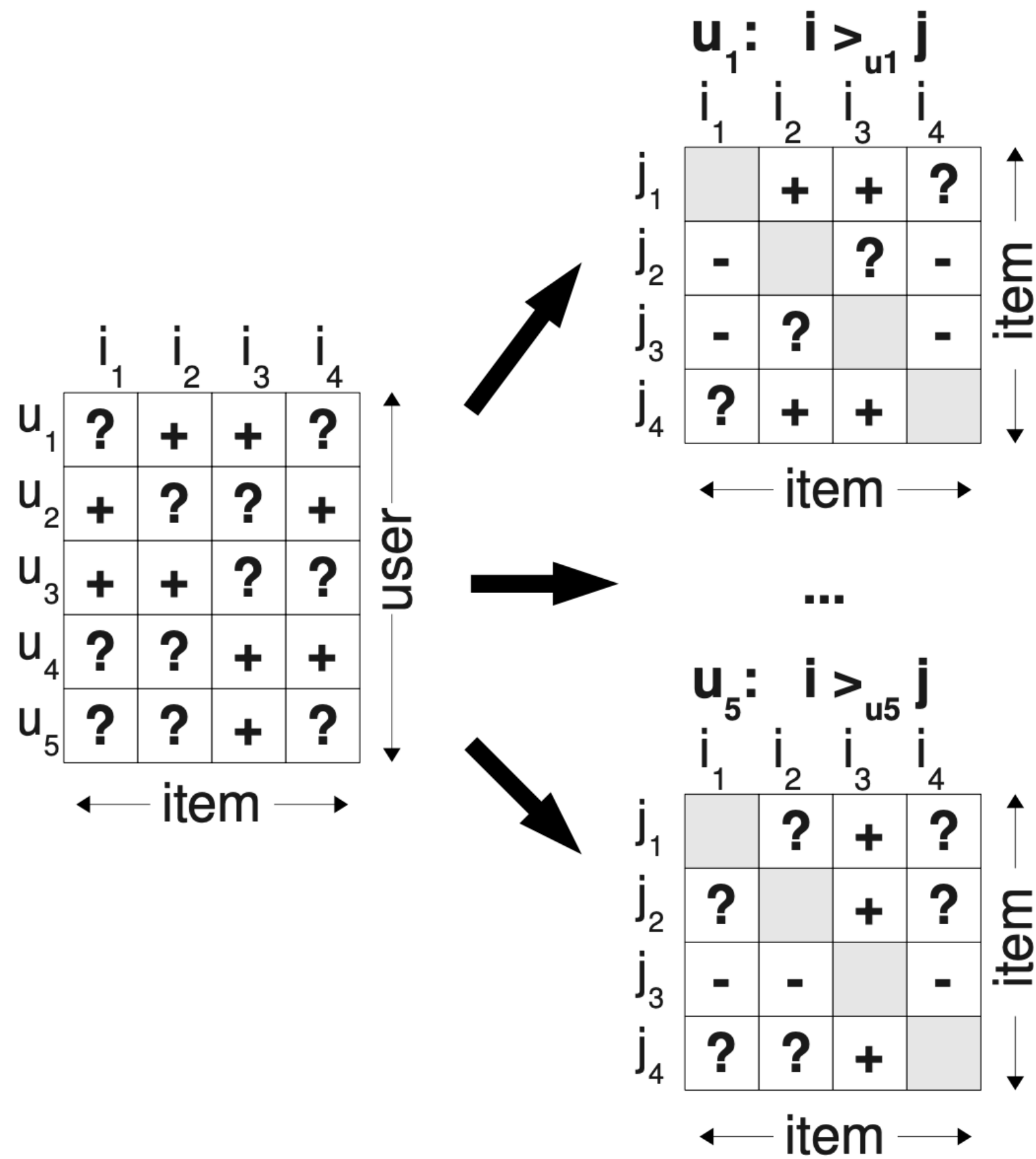
Learning to rank (L2R):

- Pointwise
- Pairwise
- Listwise

Bayesian Personalized Ranking (BPR)

- Не модель, а критерий оптимизации для pairwise L2R задачи
 - Его можно использовать для разных моделей
- Если сможем правильно сортировать любые два объекта, сможем предсказывать правильный порядок

BPR-Opt



$$D_S := \{(u, i, j) | i \in I_u^+ \wedge j \in I \setminus I_u^+\} \quad i >_u j$$

$$p(\Theta | >_u) \propto p(>_u | \Theta) p(\Theta)$$

$$\text{BPR-OPT} := \ln p(\Theta | >_u)$$

$$= \ln p(>_u | \Theta) p(\Theta)$$

$$= \ln \prod_{(u, i, j) \in D_S} \sigma(\hat{x}_{uij}) p(\Theta)$$

$$= \sum_{(u, i, j) \in D_S} \ln \sigma(\hat{x}_{uij}) + \ln p(\Theta)$$

$$= \sum_{(u, i, j) \in D_S} \ln \sigma(\hat{x}_{uij}) - \lambda_{\Theta} ||\Theta||^2$$

$$\hat{x}_{uij} := \hat{x}_{ui} - \hat{x}_{uj}$$

Heterogeneous User Actions

- User-item interactions are rarely limited to a single type of actions
- Actions of multiple types can be jointly used to build more accurate user models

Challenges:

- Not all action types are relevant for the prediction of a target action
- The elicitation of predictive action types for a target action type is done manually by using different heuristics
- There might be correlations between actions of different types

Multi-action Matrix Factorization (MMF)

Indicator function

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

Confidence function

$$c_{ui} = \begin{cases} 1 + \alpha r_{ui} & r_{ui} > 0 \\ 1 & \text{otherwise} \end{cases}$$

The target action prediction is computed using MF

$$\hat{p}_{ui} = x_u^T y_i$$

$x_u \in \mathbb{R}^f, y_i \in \mathbb{R}^f$ f -dimensional user and item latent features vectors

$$\min_{x^*, y^*} \sum_{u, i} c_{ui} (p_{ui} - x_u^T y_i)^2 + \lambda \left(\sum_u \|x_u\|^2 + \sum_i \|y_i\|^2 \right)$$

Multi-action Matrix Factorization (MMF)

Indicator function

$$p_{ui} = p(h_{ui})$$

Confidence function

$$c_{ui} = c(h_{ui})$$

The target action prediction is computed using MF

$$\hat{p}_{ui} = x_u^T y_i$$

$x_u \in \mathbb{R}^f, y_i \in \mathbb{R}^f$ f -dimensional user and item latent features vectors

$$\min_{x^*, y^*} \sum_{u, i} c_{ui} (p_{ui} - x_u^T y_i)^2 + \lambda \left(\sum_u \|x_u\|^2 + \sum_i \|y_i\|^2 \right)$$

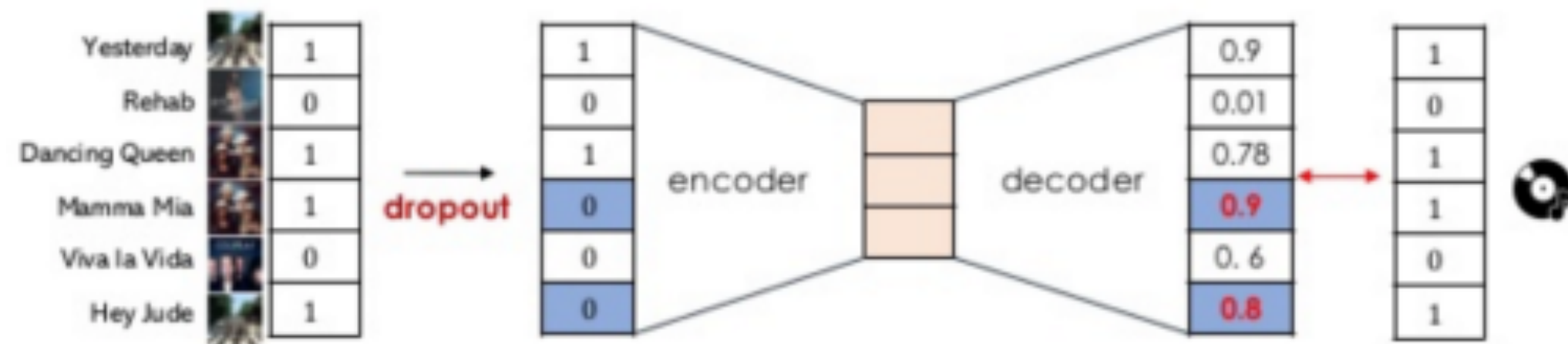
MF as a Neural Network (NN)

- Collaborative Autoencoder
- Deep MF

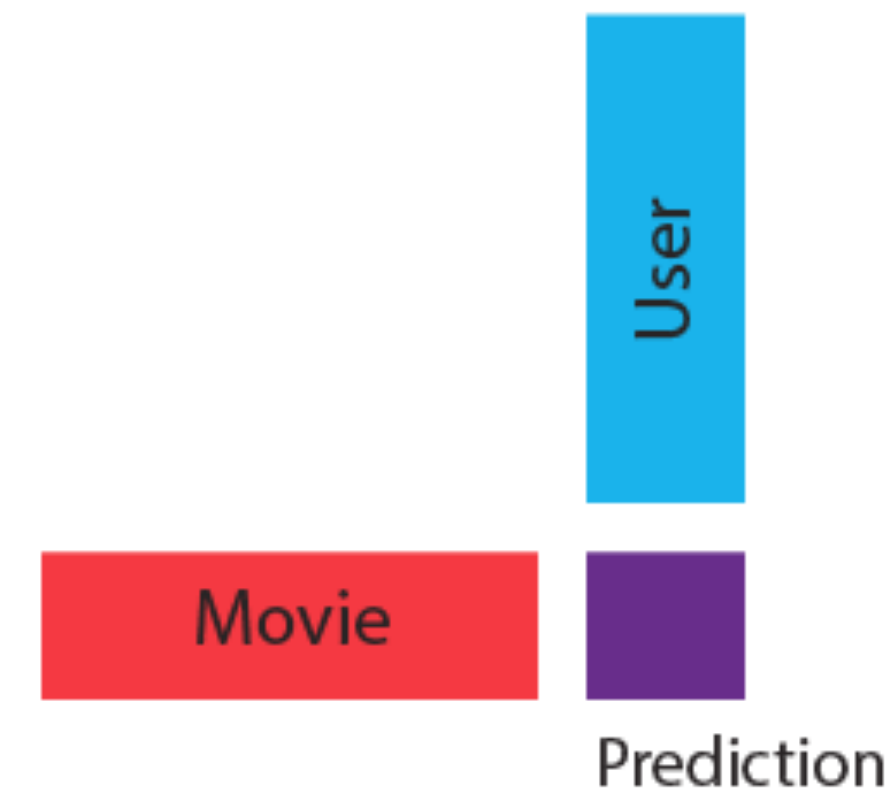
Collaborative Autoencoder

➤ Training with dropout

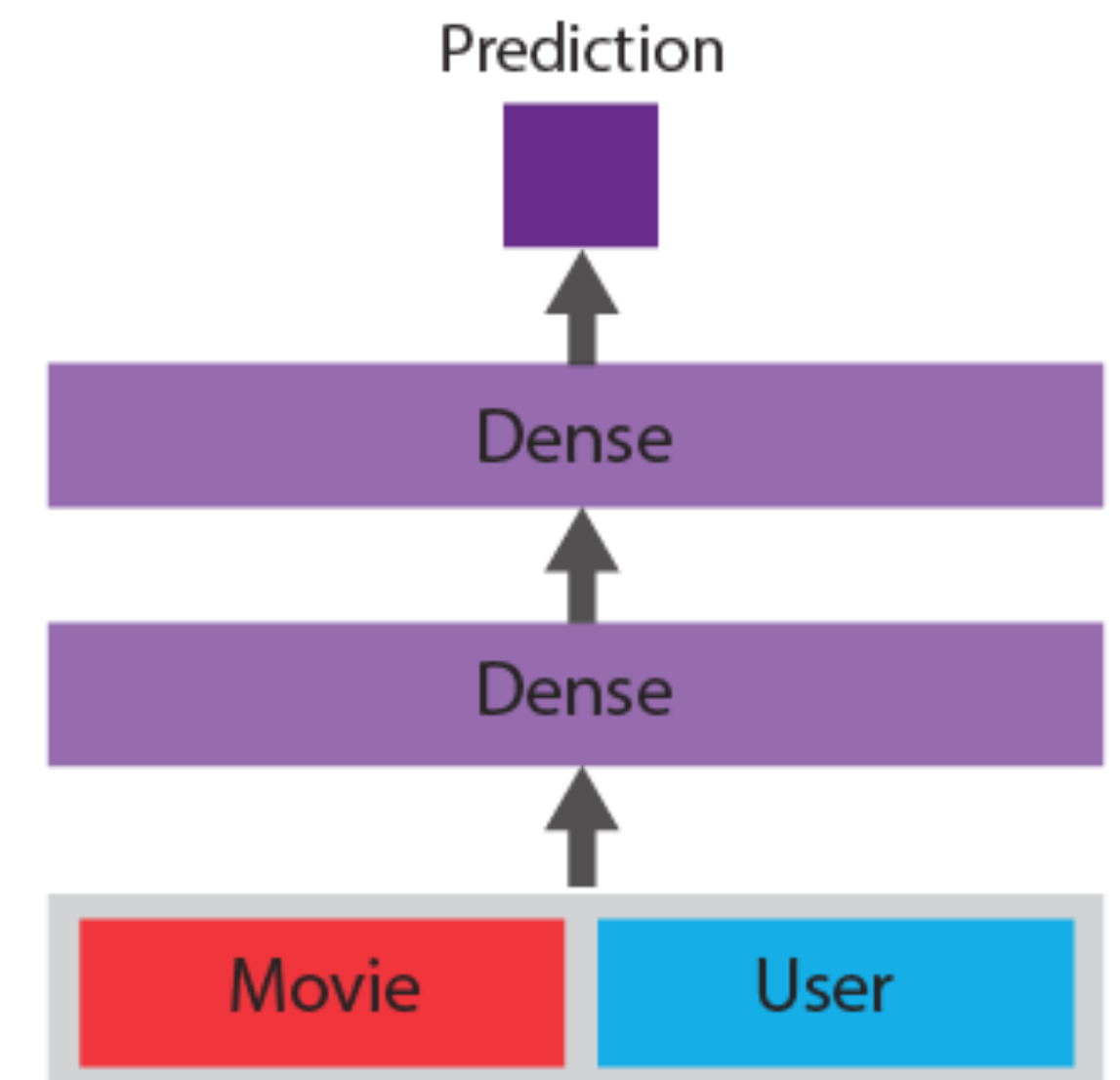
- Some positive input values are corrupted (set to zero).



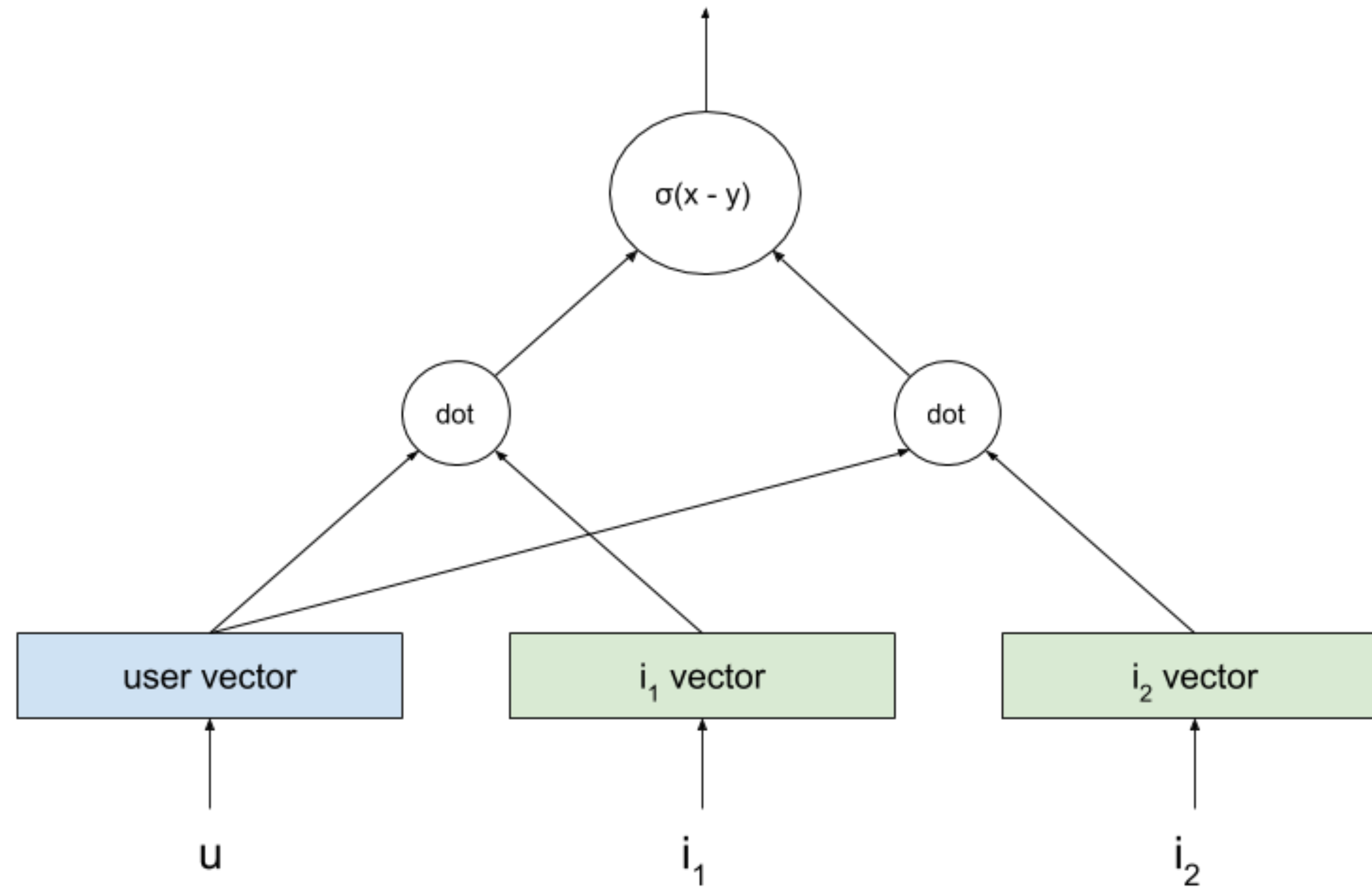
Matrix Factorization



Deep Matrix Factorization

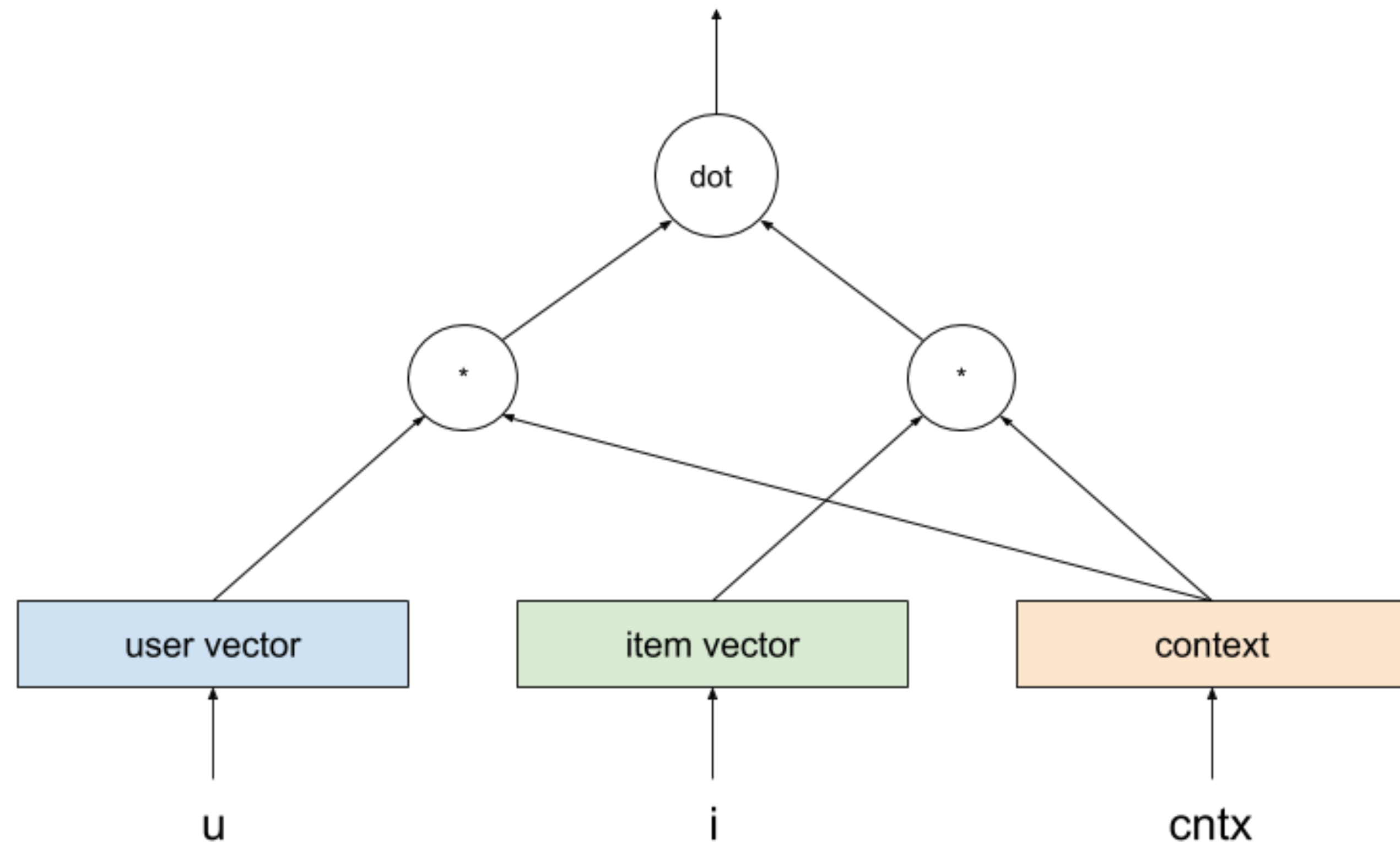


BPR as a NN



Auxiliary Features

- User and item features
- Contextual features



Wrap-up

Вспомогательный материал

Полезные ссылки

- Item-based CF: <https://dl.acm.org/doi/10.1145/963770.963776>
- WR-MF: <http://yifanhu.net/PUB/cf.pdf>
- Fold-in: <https://dl.acm.org/doi/10.1145/1454008.1454047>
- BPR-Opt: <https://arxiv.org/ftp/arxiv/papers/1205/1205.2618.pdf>
- Deep MF: <https://www.oreilly.com/content/deep-matrix-factorization-using-apache-mxnet/>
- Heterogeneous User Actions in Recommender Systems: <https://bia.unibz.it/handle/10863/13343>

Спасибо за внимание!

Вопросы ?