Коллаборативная фильтрация

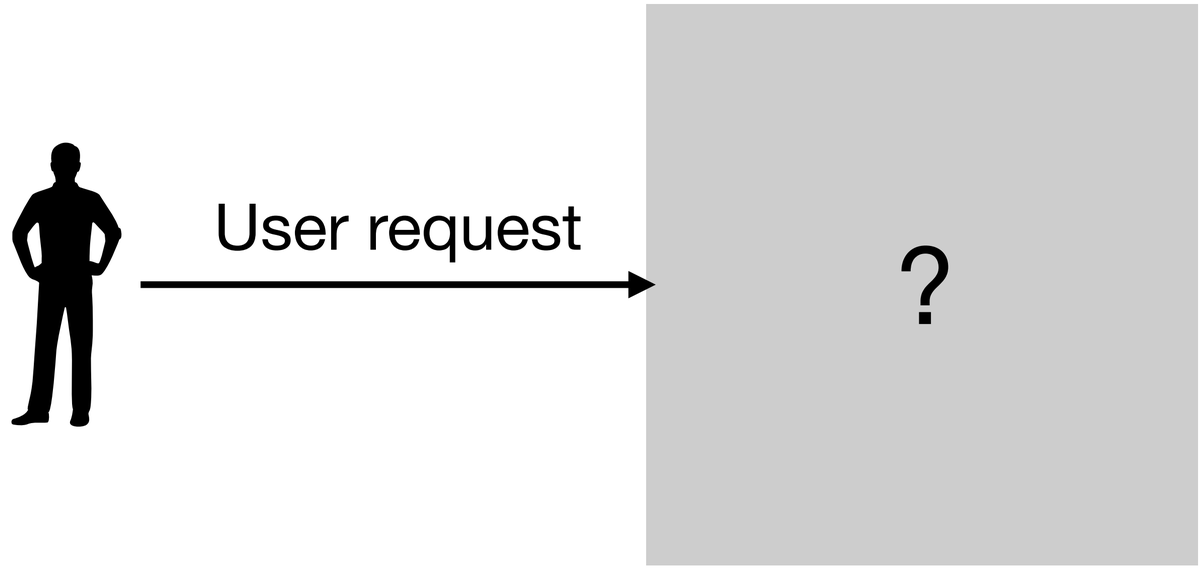
Осиновсков Илья, 10.03.2025. AI masters.

Введение

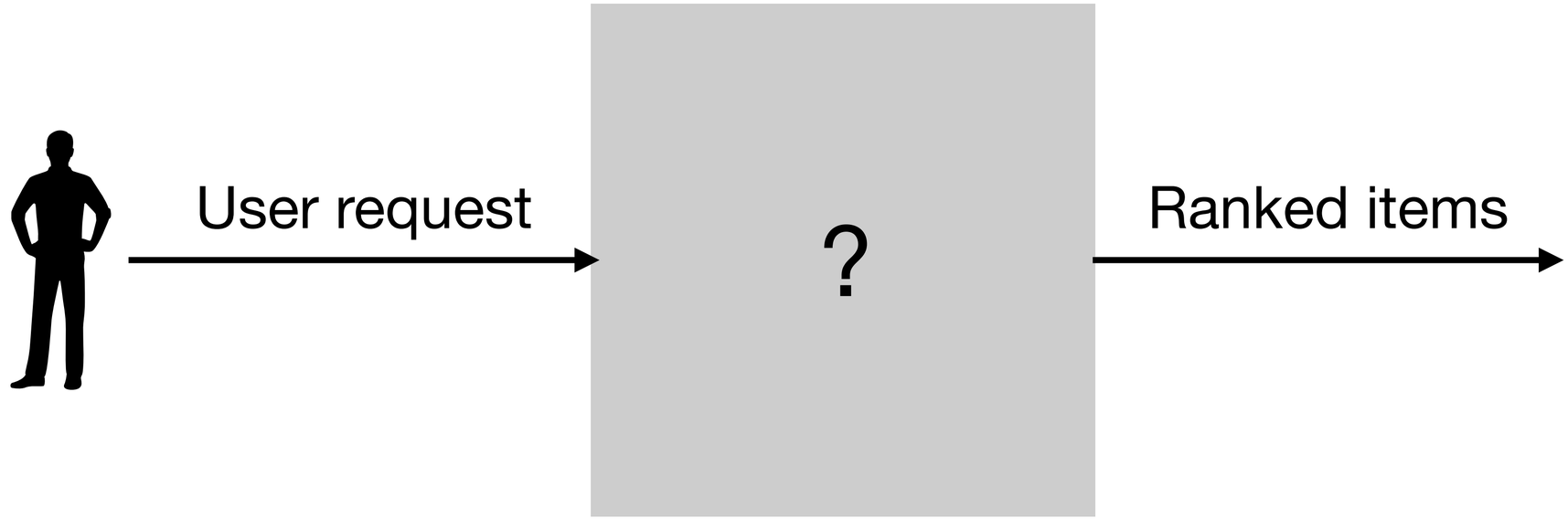
Ранжирующая система

?

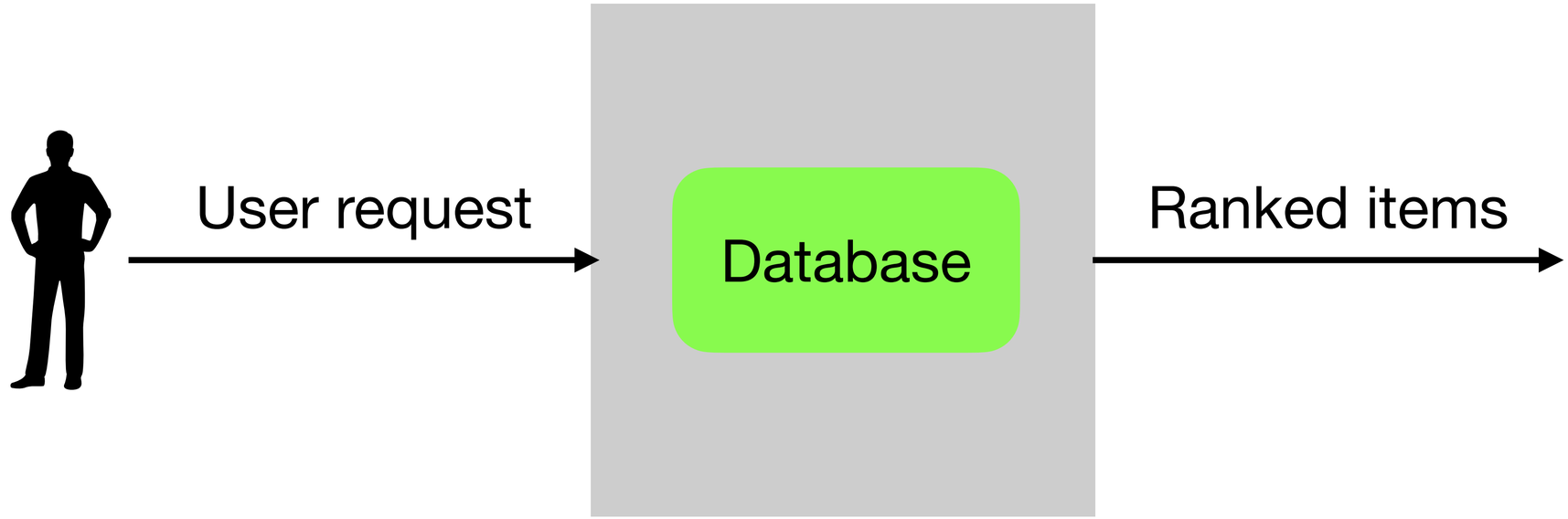
Ранжирующая система



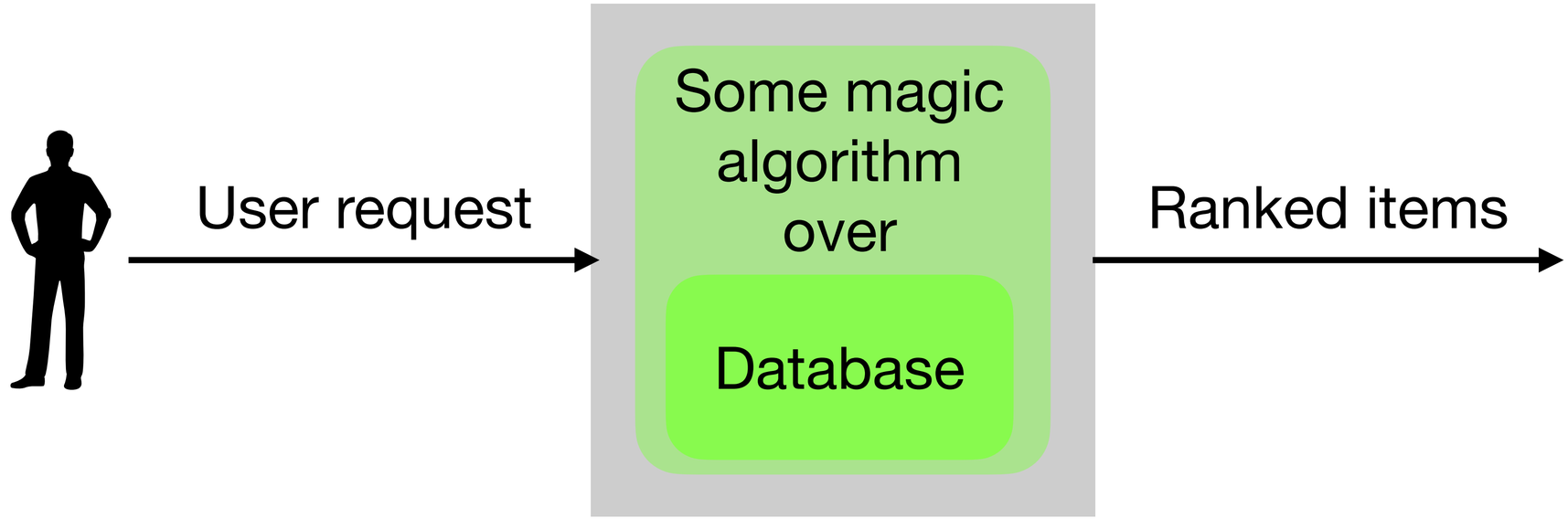
Ранжирующая система



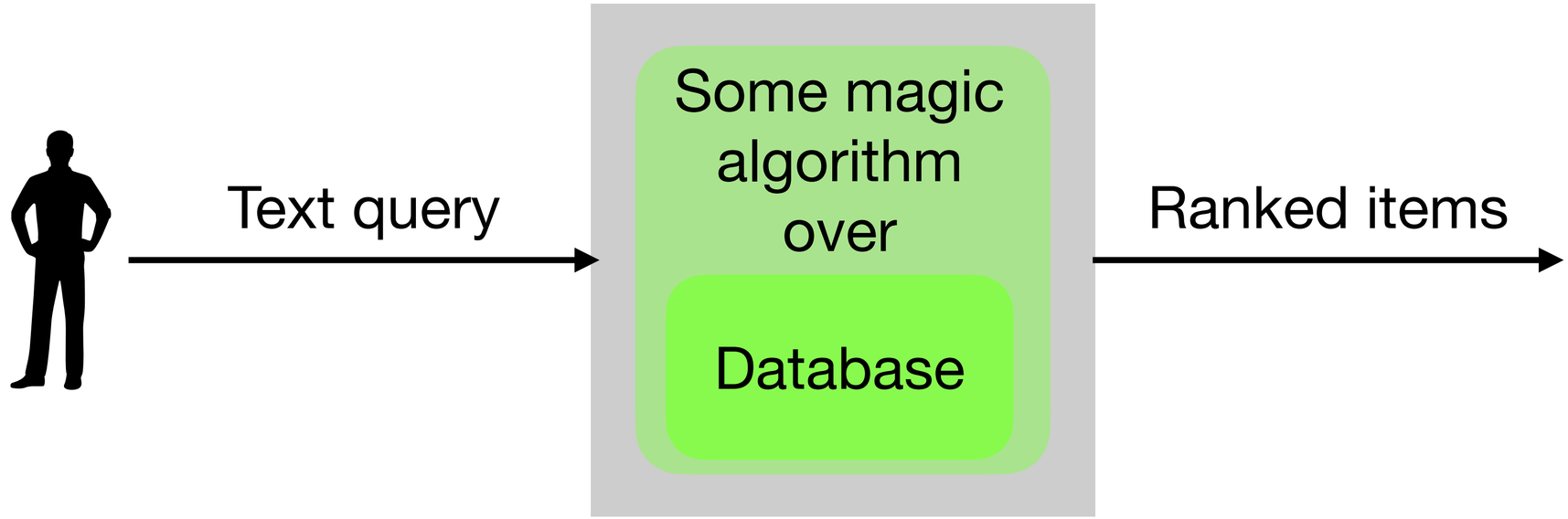
Ранжирующая система



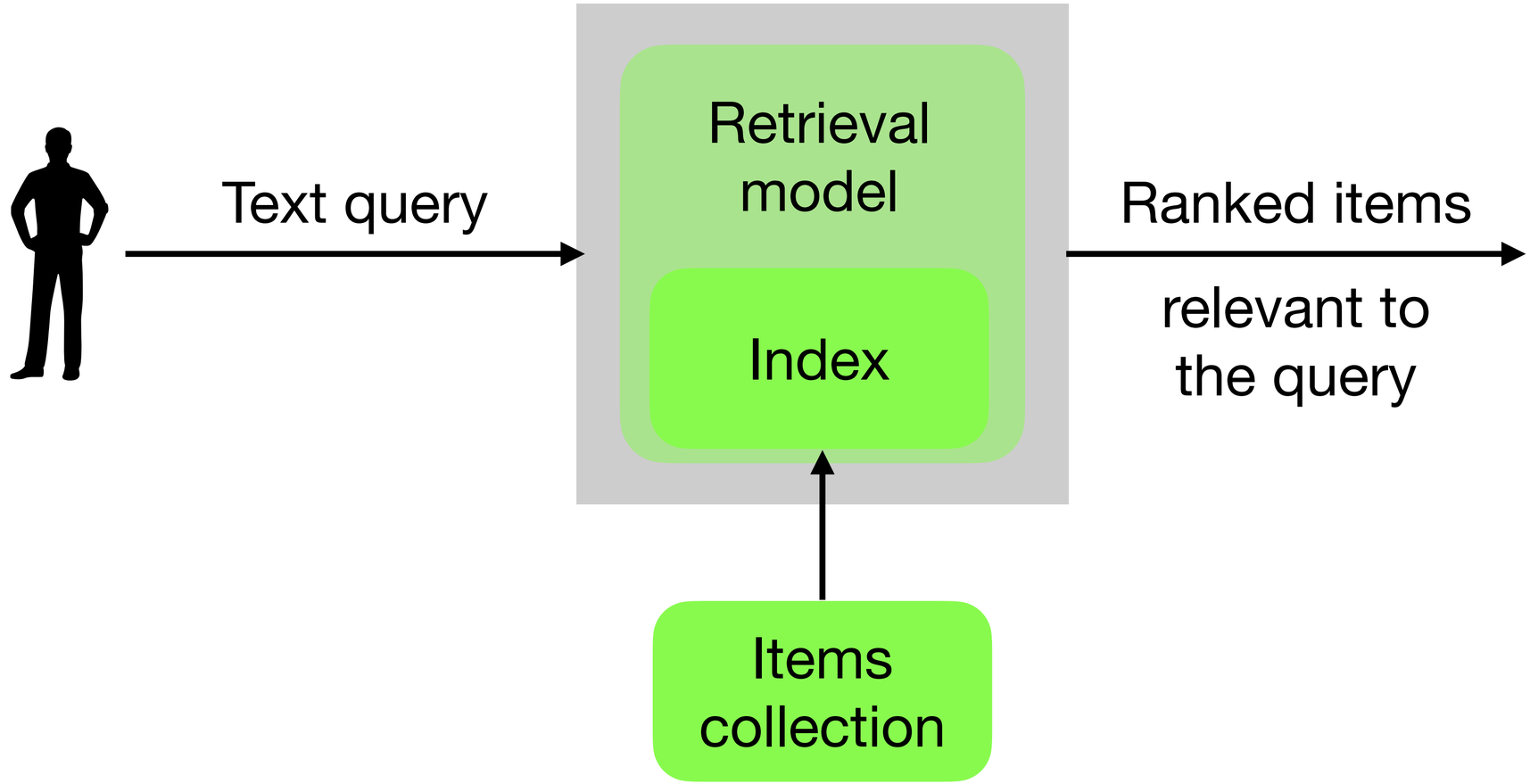
Ранжирующая система



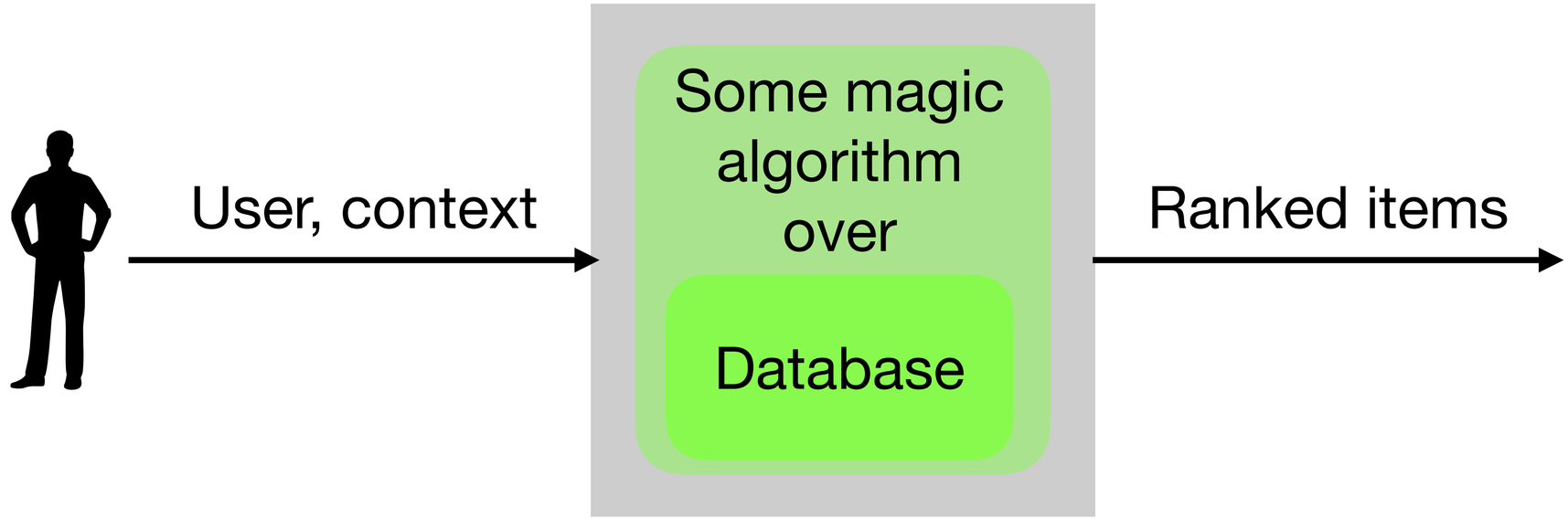
Поисковая система



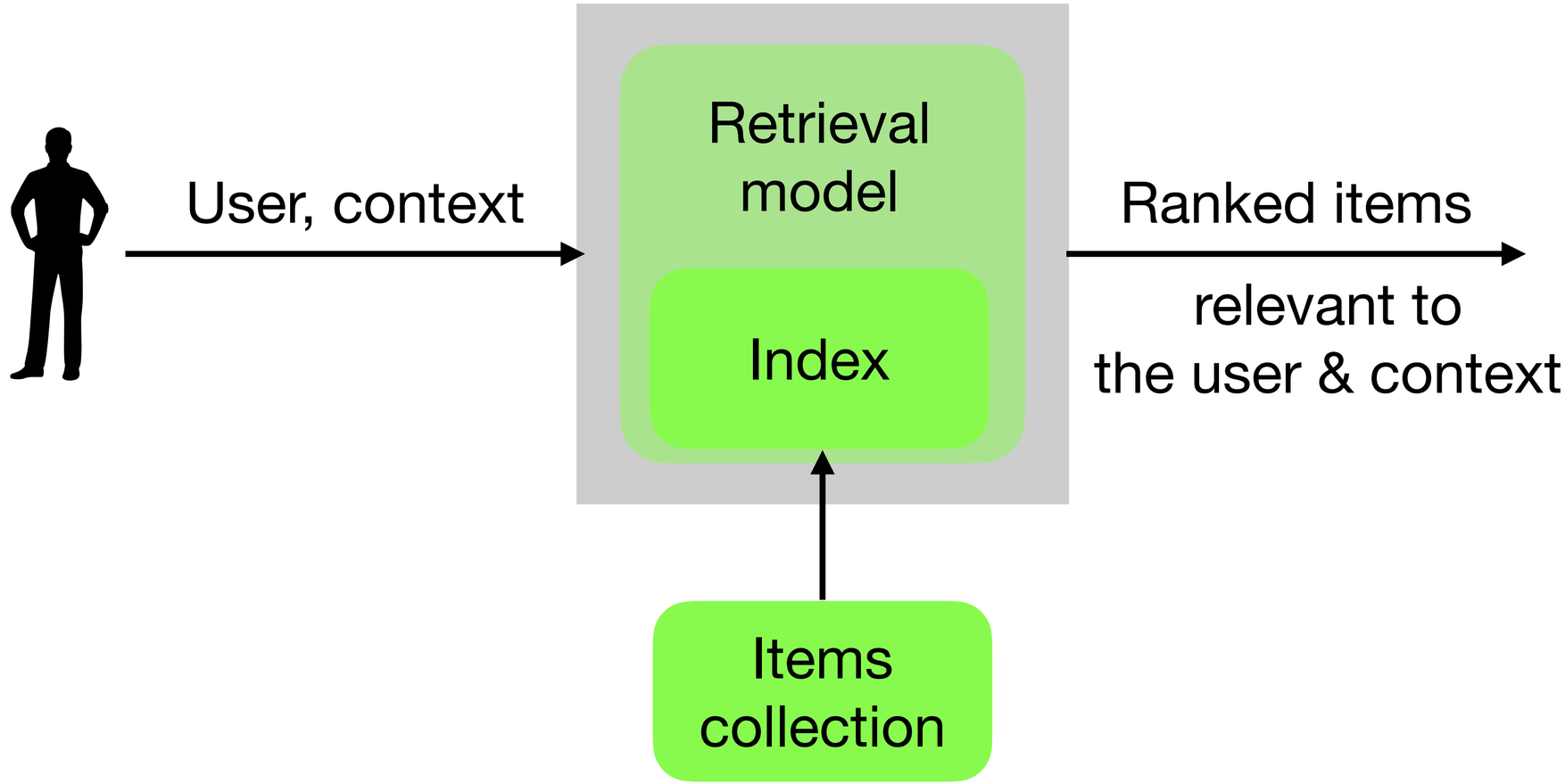
Поисковая система



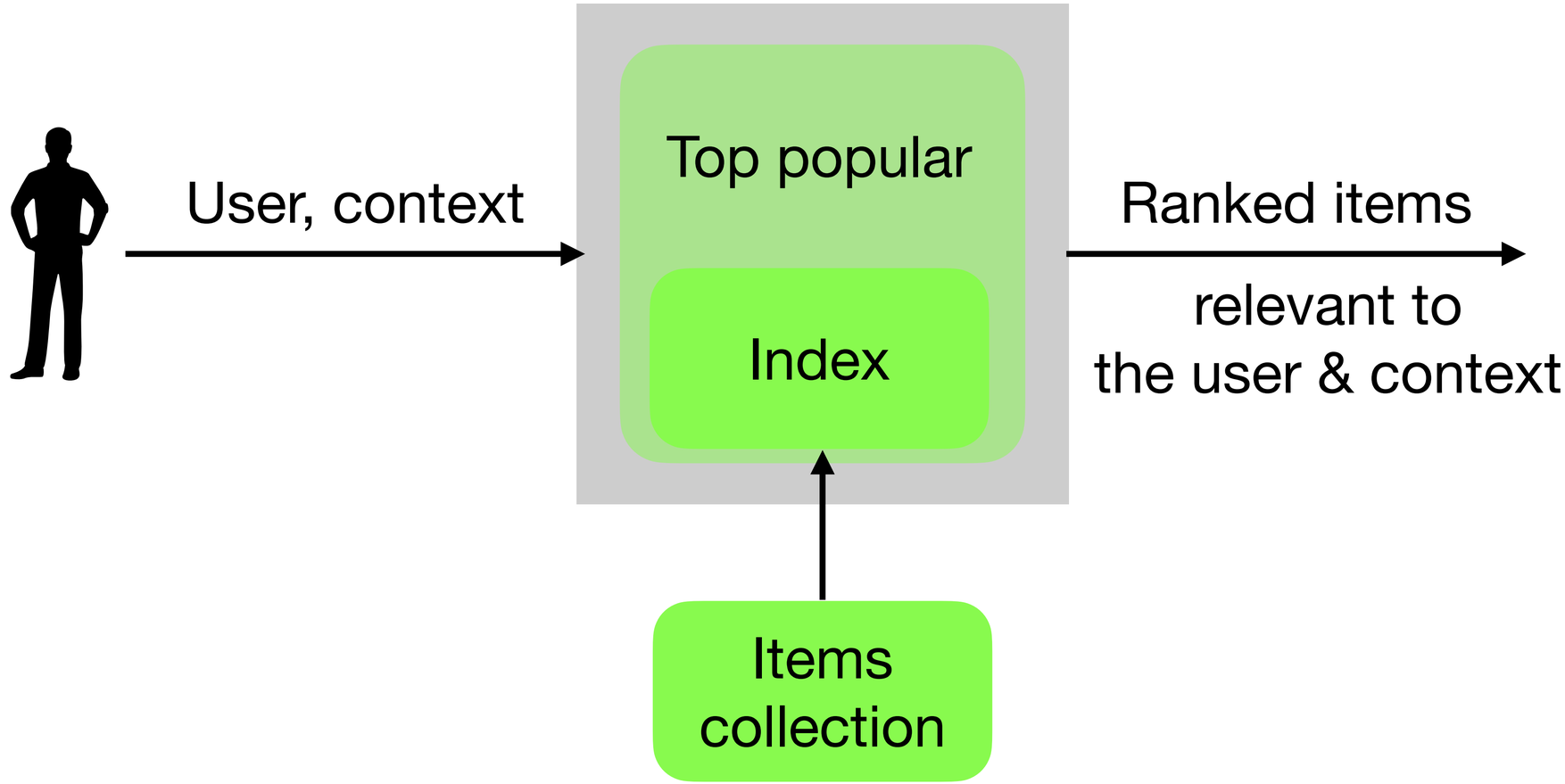
Рекомендательная система



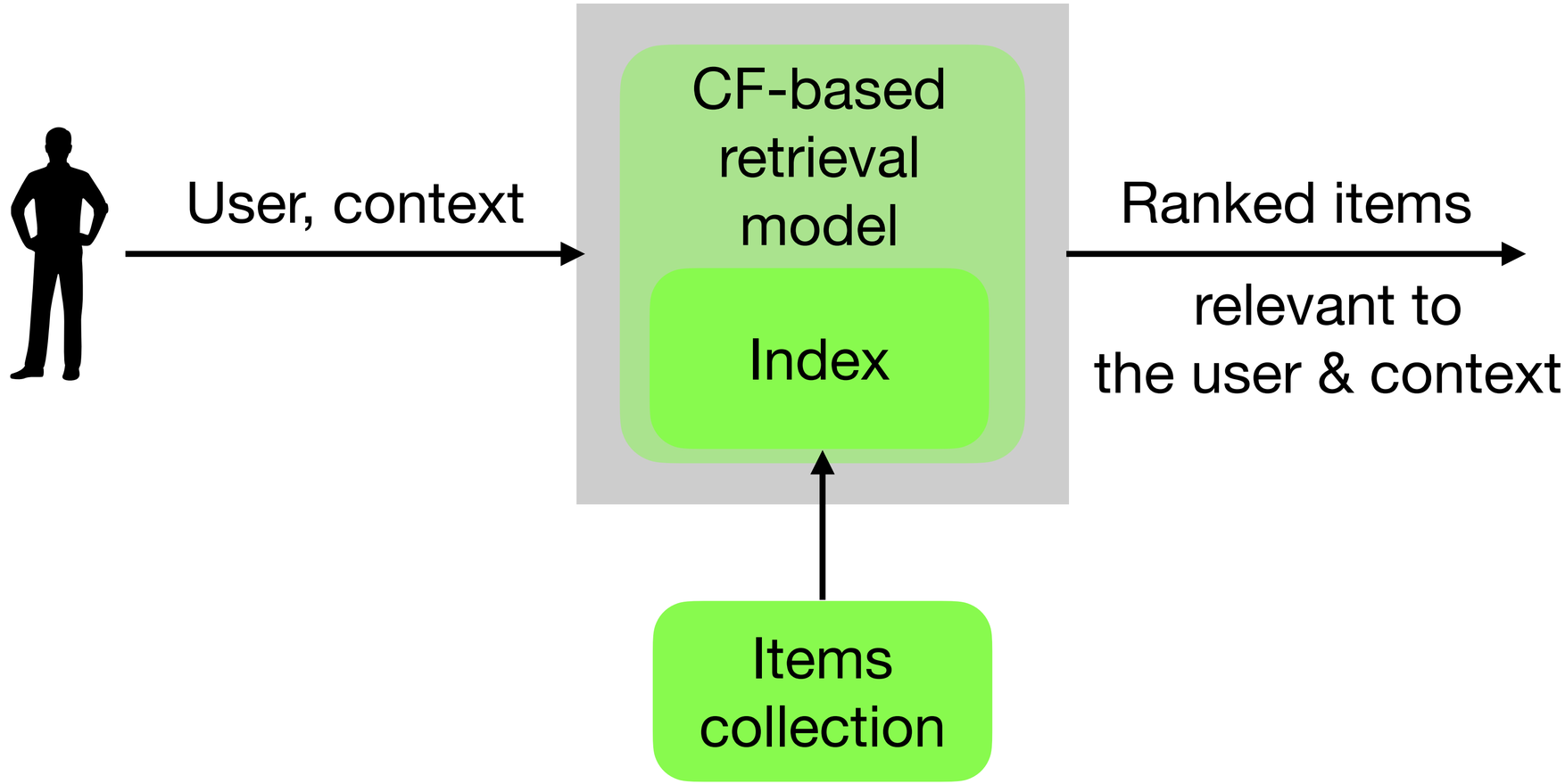
Рекомендательная система

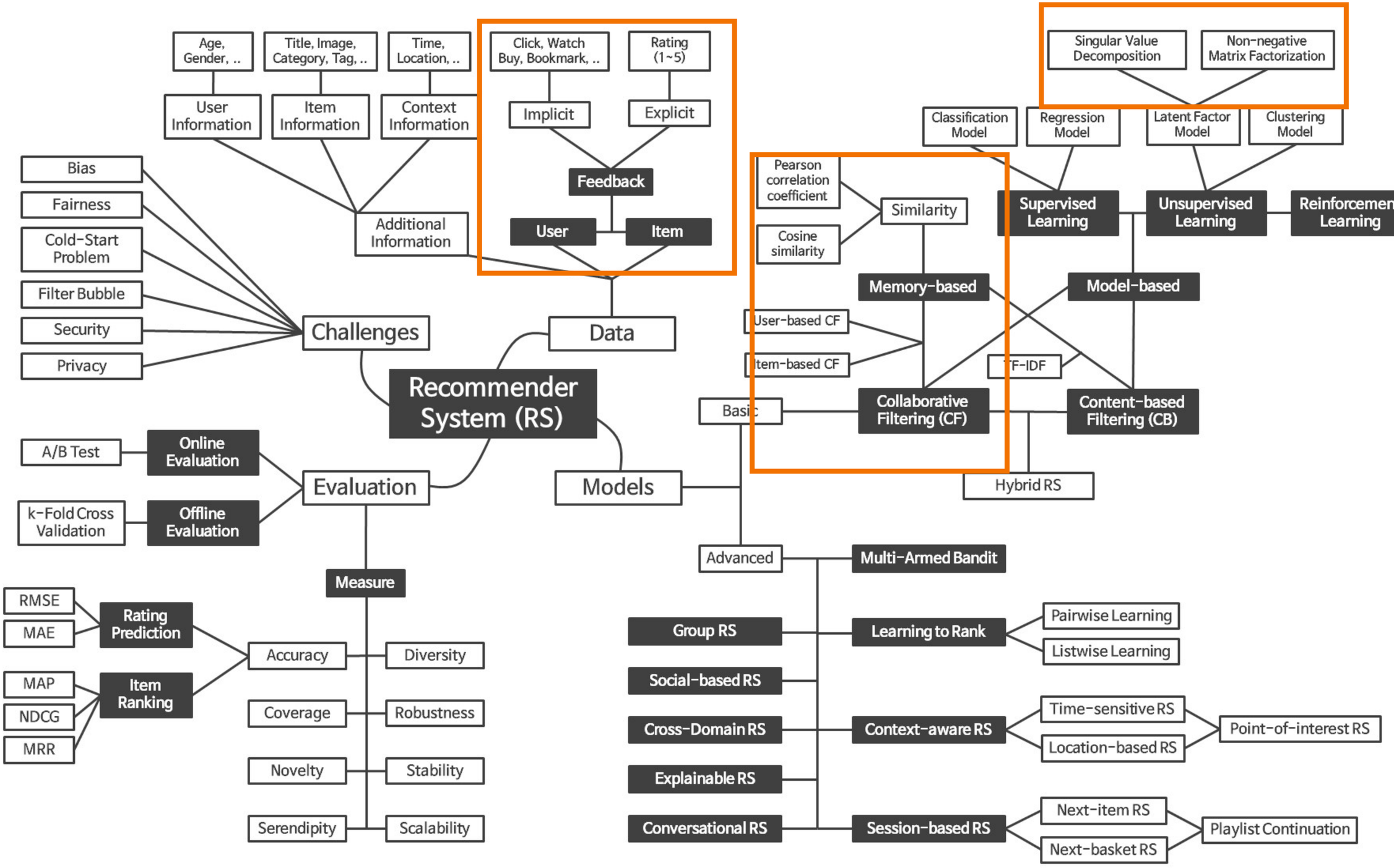


Рекомендательная система



Рекомендательная система




Recommender System (RS)
Challenges
Bias
Fairness
Cold-Start Problem
Filter Bubble
Security
Privacy
Data
Additional Information
User Information
Item Information
Context Information
Age, Gender, ..
Title, Image, Category, Tag, ..
Time, Location, ..
Feedback
Implicit
Explicit
User
Item
Click, Watch Buy, Bookmark, ..
Rating (1~5)
Evaluation
Online Evaluation
A/B Test
Offline Evaluation
k-Fold Cross Validation
Measure
Rating Prediction
RMSE
MAE
Item Ranking
MAP
NDCG
MRR
Accuracy
Diversity
Coverage
Robustness
Novelty
Stability
Serendipity
Scalability
Models
Basic
Collaborative Filtering (CF)
Similarity
Pearson correlation coefficient
Cosine similarity
Memory-based
User-based CF
Item-based CF
Model-based
Supervised Learning
Classification Model
Regression Model
Unsupervised Learning
Latent Factor Model
Clustering Model
Singular Value Decomposition
Non-negative Matrix Factorization
Reinforcement Learning
Content-based Filtering (CB)
TF-IDF
Hybrid RS
Advanced
Multi-Armed Bandit
Group RS
Learning to Rank
Pairwise Learning
Listwise Learning
Social-based RS
Cross-Domain RS
Context-aware RS
Time-sensitive RS
Location-based RS
Point-of-interest RS
Explainable RS
Conversational RS
Session-based RS
Next-item RS
Next-basket RS
Playlist Continuation

Определения и обозначения

- Типы исходных данных
- Задачи

Определения и обозначения

U - множество пользователей (users)
I - множество объектов (items)

Типы исходных данных:

- $D = (u_t, i_t, y_t)_{t=1}^{T} \in U \times I \times Y$ – транзакционные данные,
  Y – пространство описаний транзакций (фидбека)

- $R = (r_{ui})_{U \times I}$ – матрица отношений, $r_{ui} = \text{aggr}\{(u_t, i_t, y_t) \in D \mid u_t = u, i_t = i\}$
  Обычно крайне разреженная

  $r_{ui} \in \{0,1\}$ — бинарные
  $r_{ui} \in \{1,2,\ldots, M\}$ — рейтинги (порядковые или целые)

Определения и обозначения

Приведем несколько примеров фидбека:

- Для товара – факт добавления в корзину;
- Для музыки – дослушали ли трек до конца;
- Для статьи – лайк/дизлайк;
- Для видео – время его просмотра или факт просмотра, например, наполовину.

Определения и обозначения

Типы фидбека

- Explicit (явный фидбек)
- Implicit (неявный фидбек)

Определения и обозначения

Типы фидбека

- Explicit (явный фидбек)
  Действия пользователя, по которым точно можно понять, понравился ли ему объект.
  Например: лайк/дизлайк, отзыв с рейтингом.

- Implicit (неявный фидбек)

Определения и обозначения

Типы фидбека

- Explicit (явный фидбек)
- Implicit (неявный фидбек)
  Любая другая информация о действиях пользователя. Выступает прокси к явному фидбеку
  Например: клик по ссылке, покупка товара, просмотр видео больше X% по длительности.

Определения и обозначения

Типы фидбека

- Explicit (явный фидбек)
- Implicit (неявный фидбек)
  Любая другая информация о действиях пользователя. Выступает прокси к явному фидбеку
  Например: клик по ссылке, покупка товара, просмотр видео больше X% по длительности.

  Неявного фидбека обычно сильно больше чем явного, но он шумный.

  Из-за различий для каждого фидбека есть разные техники обработки и использования, которые будут обсуждаться далее.

Определения и обозначения

U - множество пользователей (users)
I - множество объектов (items)

Типы исходных данных:

- $D = (u_t, i_t, y_t)_{t=1}^{T} \in U \times I \times Y$ – транзакционные данные,
  Y – пространство описаний транзакций (фидбека)

- $R = (r_{ui})_{U \times I}$ – матрица отношений, $r_{ui} = \text{aggr}\{(u_t, i_t, y_t) \in D \mid u_t = u, i_t = i\}$
  Обычно крайне разреженная

  $r_{ui} \in \{0,1\}$ — бинарные
  $r_{ui} \in \{1,2,\ldots, M\}$ — рейтинги (порядковые или целые)

Определения и обозначения

Задачи в рекомендательных системах:

- формирование списка рекомендаций для u или для i
- оценивание сходства: $\rho(u, u'), \rho(i, i'), \rho(u, i)$
- заполнение пропусков в ячейках r_{ui}

Пример 1

U - клиенты интернет магазина
I - товары
$r_{ui} =$ [клиент u посмотрел/купил товар i]

Задачи персонализации предложений:

- формирование списка рекомендаций для u
- заполнение пропусков в ячейках r_{ui}

Пример 2

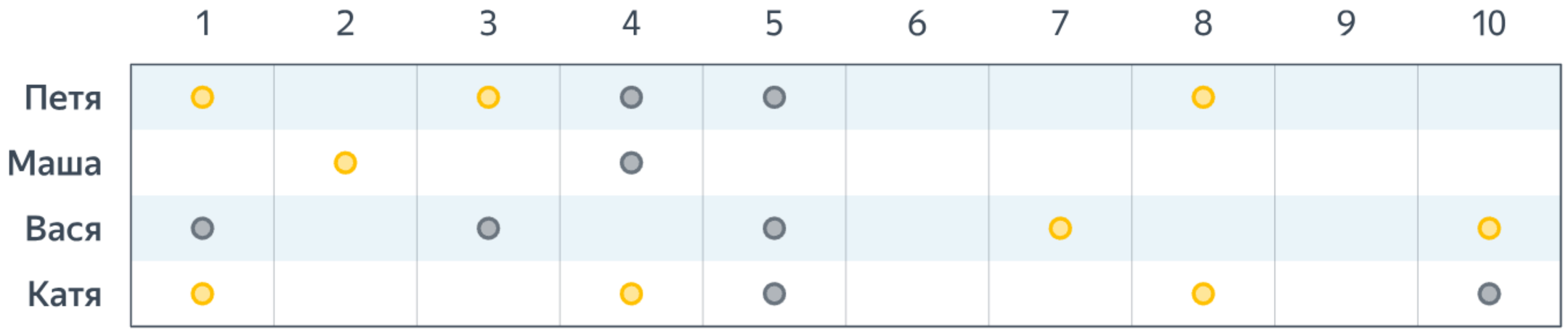
U - клиенты онлайн кинотеатра
I - фильмы
$r_{ui} =$ рейтинг, который клиент u поставил фильму i

Конкурс Netflix. Датасет

- 2 октября 2006 - 21 сентября 2009
- приз 1_000_000$
- |U| = 480189 фильмов, |I| = 17770 клиентов
- 10^8 рейтингов фильмов по шкале от 1 до 5
- Точность прогнозов оценивалась по тестовой выборке $\text{RMSE}^2 = \frac{1}{|D'|} \sum_{(u,i) \in D'} (r_{ui} - \hat{r}_{ui})^2$
- Задача: уменьшить RMSE с 0.9514 до 0.8563 (на 10%)

Коллаборативная фильтрация

Что рекомендуем Кате?



Коллаборативная фильтрация

Семейство методов рекомендаций, использующих сходство по истории взаимодействия между пользователем и объектом.

Основные подходы

- Корреляционные модели (Memory-Based CF)
- Латентные семантические модели (Latent Model-Based CF)

Основные подходы

- Корреляционные модели (Memory-Based CF)
  - хранение всей матрицы R
  - сходство пользователей - корреляция строк матрицы R
  - сходство объектов - корреляция столбцов матрицы R
- Латентные семантические модели (Latent Model-Based CF)

User/Item-KNN

User-based CF

- Оценка рейтинга на основе рейтингов похожих пользователей

$$\hat{r}_{ui} = \bar{r}_u + \frac{\sum_{v \in U(u)} S(u,v)(r_{vi} - \bar{r}_v)}{\sum_{v \in U(u)} S(u,v)}$$

где

$U(u)$ – множество похожих на u пользователей (коллаборация)

$\bar{r}_u$ – средний рейтинг клиента u

$S(u,v)$ – функция похожести пары клиентов u и v

User-based CF

- Предположение о разном среднем рейтинге

Item-based CF

- Оценка рейтинга на основе рейтингов похожих айтемов

$$\hat{r}_{ui} = \bar{r}_i + \frac{\sum_{j \in I(i)} S(i,j)(r_{ui} - \bar{r}_j)}{\sum_{j \in I(i)} S(i,j)}$$

где

$I(i)$ – множество похожих на i айтемов

$\bar{r}_i$ – средний рейтинг айтема i

$S(i,j)$ – функция похожести пары айтемов i и j

Функции похожести

- Корреляция Пирсона
- Косинусная похожесть
- Мера Жаккара
- etc (76+ вариантов похожести)

Функции похожести

Корреляция Пирсона

- $$S(u,v) = \frac{\sum_{i \in I(u,v)} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in I(u,v)} (r_{ui} - \bar{r}_u)^2} \cdot \sqrt{\sum_{i \in I(u,v)} (r_{vi} - \bar{r}_v)^2}}$$

  Здесь $I(u)$ — множество объектов, которые клиент u оценил, и $I(u,v)$ — множество объектов, которые оценили оба клиента u и v.

- Функция сходства $S(i,j)$ для пар объектов определяется аналогично.

Функции похожести

Косинусная мера сходства

- $$S(u,v) = \frac{\sum_{i \in I(u,v)} r_{ui} r_{vi}}{\sqrt{\sum_{i \in I(u)} r_{ui}^2} \cdot \sqrt{\sum_{i \in I(v)} r_{vi}^2}}$$

  Здесь подразумевается, что $r_{ui} = 0$, если $i \neq I(u)$.

- Функция сходства $S(i,j)$ для пар объектов определяется аналогично.

Функции похожести

Мера (коэффициент) Жаккара

- Определяет пересечение 2 наборов объектов/пользователей A и B:

$$Jaccard(A, B) = \frac{A \cap B}{A \cup B}$$

- Объекты/пользователи представляются как мешок слов (BoW)
- $S(u, v) = Jaccard(u, v)$

Функции похожести

Мера (коэффициент) Жаккара

- Определяет пересечение 2 наборов объектов/пользователей A и B:

$$Jaccard(A, B) = \frac{A \cap B}{A \cup B}$$

- Пример:
  - u_1 = {1, 2, 3, 4, 5, 6}
  - u_2 = {5, 6, 7, 8, 9, 10}
  - $S(u_1, u_2) = Jaccard(u_1, u_2) = \frac{2}{10} = 0.2$

User/item сравнение

Table 2.2 The space and time complexity of user-based and item-based neighborhood methods, as a function of the maximum number of ratings per user $p = \max_u |\mathcal{I}_u|$, the maximum number of ratings per item $q = \max_i |\mathcal{U}_i|$, and the maximum number of neighbors used in the rating predictions k

	Space	Time	
		Training	Online
User-based	$O(\|\mathcal{U}\|^2)$	$O(\|\mathcal{U}\|^2 p)$	$O(\|\mathcal{I}\|k)$
Item-based	$O(\|\mathcal{I}\|^2)$	$O(\|\mathcal{I}\|^2 q)$	$O(\|\mathcal{I}\|k)$

Насколько это актуально?

Top-N Recommendation Algorithms: A Quest for the State-of-the-Art

VITO WALTER ANELLI, Politecnico di Bari, Italy
ALEJANDRO BELLOGÍN, Universidad Autónoma de Madrid, Spain
TOMMASO DI NOIA, Politecnico di Bari, Italy
DIETMAR JANNACH, University of Klagenfurt, Austria
CLAUDIO POMO, Politecnico di Bari, Italy

Rank	Algorithm	Count
1	$EASE^R$	185
2	$RP^3\beta$	169
3	SLIM	160
4	UserKNN	154
5	MF2020	115
6	ItemKNN	99
7	MultiVAE	92
8	iALS	90
9	NeuMF	61
10	BPRMF	45
11	MostPop	18
12	Random	0

(a) Overall

Линейные модели

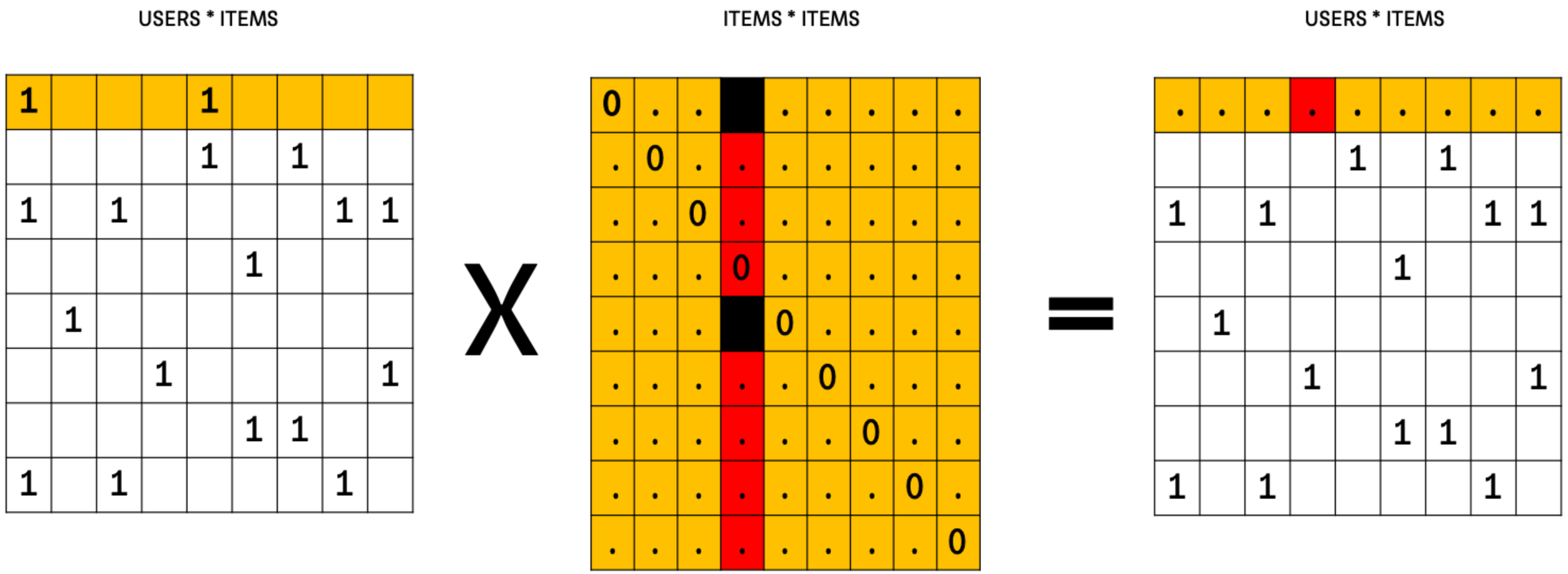
Разряженная линейная модель (SLIM, 2011)

- Item based: оцениваем неизвестные рейтинги как линейную комбинацию по известным рейтингам других объектов вместо функции похожести

Разряженная линейная модель (SLIM)

- Item based: оцениваем неизвестные рейтинги как линейную комбинацию по известным рейтингам других объектов вместо функции похожести
- $\hat{r}_{ui} = \sum_{j \in I \setminus i} r_{uj} w_{ij} = \langle r_u, w_i \rangle$
- $\hat{r}_i = R w_i, \quad i \in I$
- $\frac{1}{2}\|r_i - R w_i\|^2 + \frac{\beta}{2}\|w_i\|_2^2 + \lambda \|w_i\|_1 \to \min\limits_{w_i}$
  при условиях $(w_i > 0)$ и $(w_{ii} = 0)$

Разряженная линейная модель (SLIM)



$$\hat{r}_{ui} = \mathbf{r}_u \mathbf{w}_i^\top$$

$$\underset{W}{\text{minimize}} \quad \frac{1}{2}\|R - RW\|_F^2 + \frac{\beta}{2}\|W\|_F^2 + \lambda\|W\|_1$$

$$\text{subject to} \quad W \geq 0$$

$$\text{diag}(W) = 0.$$

EASE (2019)

Embarrassingly Shallow Autoencoders for Sparse Data*

Harald Steck
Netflix
Los Gatos, California
hsteck@netflix.com

The predicted score $S_{u,j}$ for an item $j \in \mathcal{I}$ given a user $u \in \mathcal{U}$ is defined by the dot product

$$S_{uj} = X_{u,\cdot} \cdot B_{\cdot,j}, \tag{1}$$

where $X_{u,\cdot}$ refers to row u, and $B_{\cdot,j}$ to column j.

$$\min_{B} \quad ||X - XB||_F^2 + \lambda \cdot ||B||_F^2$$
$$\text{s.t.} \quad \text{diag}(B) = 0$$

EASE

$$\underset{W}{\text{minimize}} \quad \frac{1}{2}\|R - RW\|_F^2 + \frac{\beta}{2}\|W\|_F^2 + \lambda\|W\|_1$$
$$\text{subject to} \quad W \geq 0$$
$$\text{diag}(W) = 0.$$

SLIM

EASE

Есть решение в явном виде

$$\hat{P} \triangleq (X^\top X + \lambda I)^{-1}$$

$$\hat{B}_{i,j} = \begin{cases} 0 & \text{if } \; i = j \\ -\dfrac{\hat{P}_{ij}}{\hat{P}_{jj}} & \text{otherwise.} \end{cases}$$

EASE

Преимущества и недостатки

- Преимущества
  - Есть явное решение
- Недостатки
  - Сложность $O(I^3)$ для вычисления обратной матрицы

EASE

Метрики

Task	Dataset	Model	Metric Name	Metric Value	Global Rank
Recommendation Systems	Million Song Dataset	EASE	Recall@20	0.333	# 1
			Recall@50	0.428	# 1
			nDCG@100	0.389	# 1
Recommendation Systems	MovieLens 20M	EASE	Recall@20	0.391	# 7
			Recall@50	0.521	# 7
			nDCG@100	0.420	# 6
Recommendation Systems	Netflix	EASE	Recall@20	0.362	# 2
			Recall@50	0.445	# 4
			nDCG@100	0.393	# 3

Выводы по memory-based подходам

- Преимущества
  - Интерпретируемость
  - Легкость реализации
- Недостатки
  - Надо хранить матрицу R
  - Холодный старт
  - Возможная тривиальность рекомендаций

Основные подходы

- Корреляционные модели (Memory-Based CF)
  - хранение всей матрицы R
  - сходство пользователей - корреляция строк матрицы R
  - сходство объектов - корреляция столбцов матрицы R
- Латентные семантические модели (Latent Model-Based CF)
  - расчет представлений (эмбедингов) пользователей и объектов
  - хранение эмбедингов вместо хранения R
  - сходство пользователей и объектов - близость их эмбедингов

Латентные модели

Низкоранговые разложения матриц

- SVD
- ALS
- iALS
- etc

Латентные модели

Низкоранговые разложения матриц

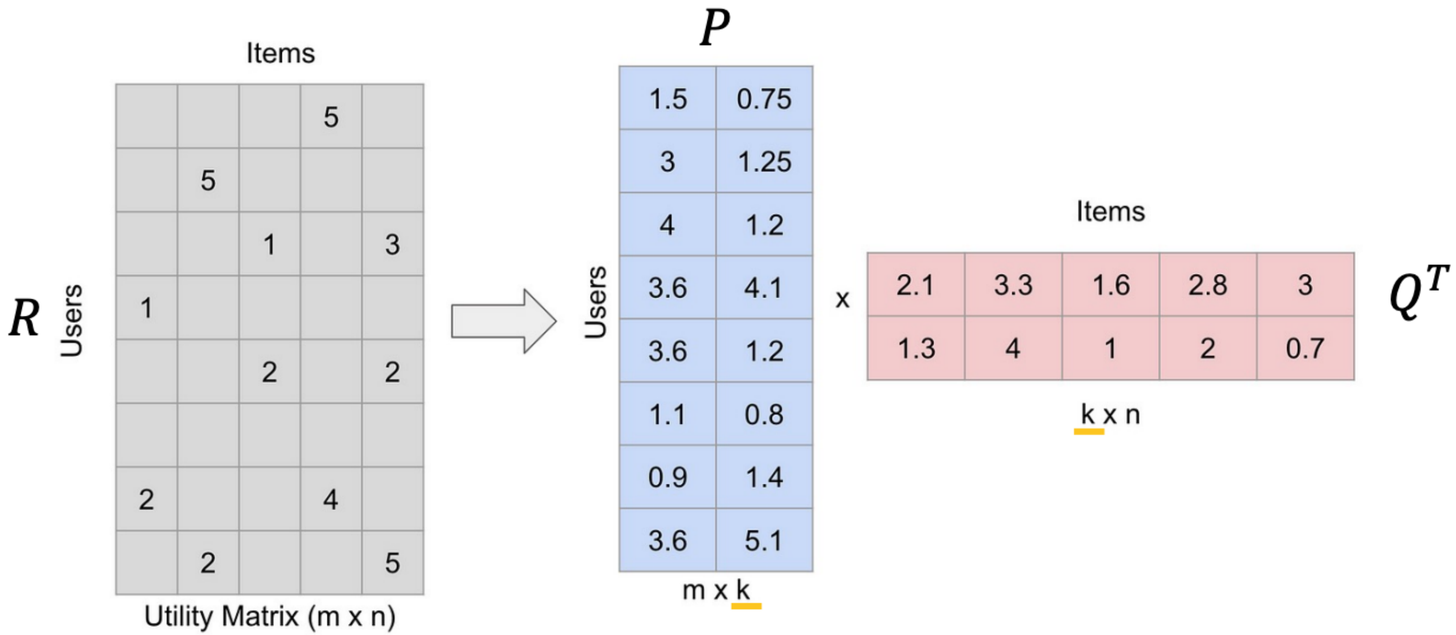
- T - множество интересов
  p_{ut} – векторное представление клиента u
  q_{it} – векторное представление объекта i
- Задача:
  Найти разложение $\hat{r}_{ui} = \sum_{t \in T} p_{ut} q_{it} = \langle p_u, q_i \rangle$

  Матричная запись: $\hat{R} = PQ^\top$, критерий $\|R - PQ^\top\| \to \min\limits_{P,Q}$
- Вероятностная интерпретация: $p(i \mid u) = \sum_{t \in T} q(i \mid t) p(t \mid u)$

Singular value decomposition

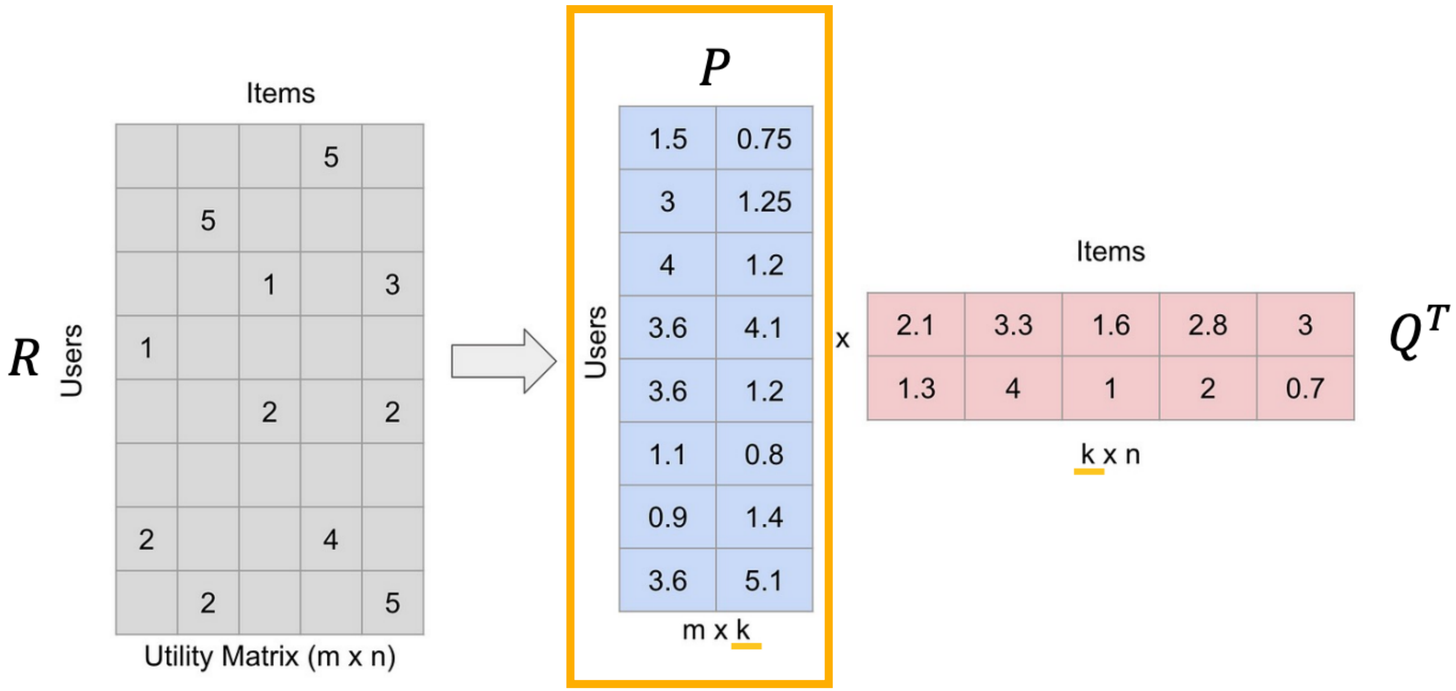
- Постановка задачи: $\|R - PQ^\top\| \to \min\limits_{P,Q}$
- $\hat{R} = V\sqrt{D}\sqrt{D}U^T, \quad U^TU = I, \quad V^TV = I$
- Достоинства:
  - Можно применять готовые библиотеки линейной алгебры
  - Хорошее ранжирование предложений на некоторых данных
- Недостатки:
  - Если r_{ui} не известно, то полагаем $r_{ui} = 0$ (неявно считаем, что если клиент i никогда не выбирал объект, то он ему, скорее всего, не интересен)
  - Ортогональность (собственных) векторов p_t, q_t
  - Неинтерпретируемость компонент векторов p_u, q_i

Alternating Least Squares



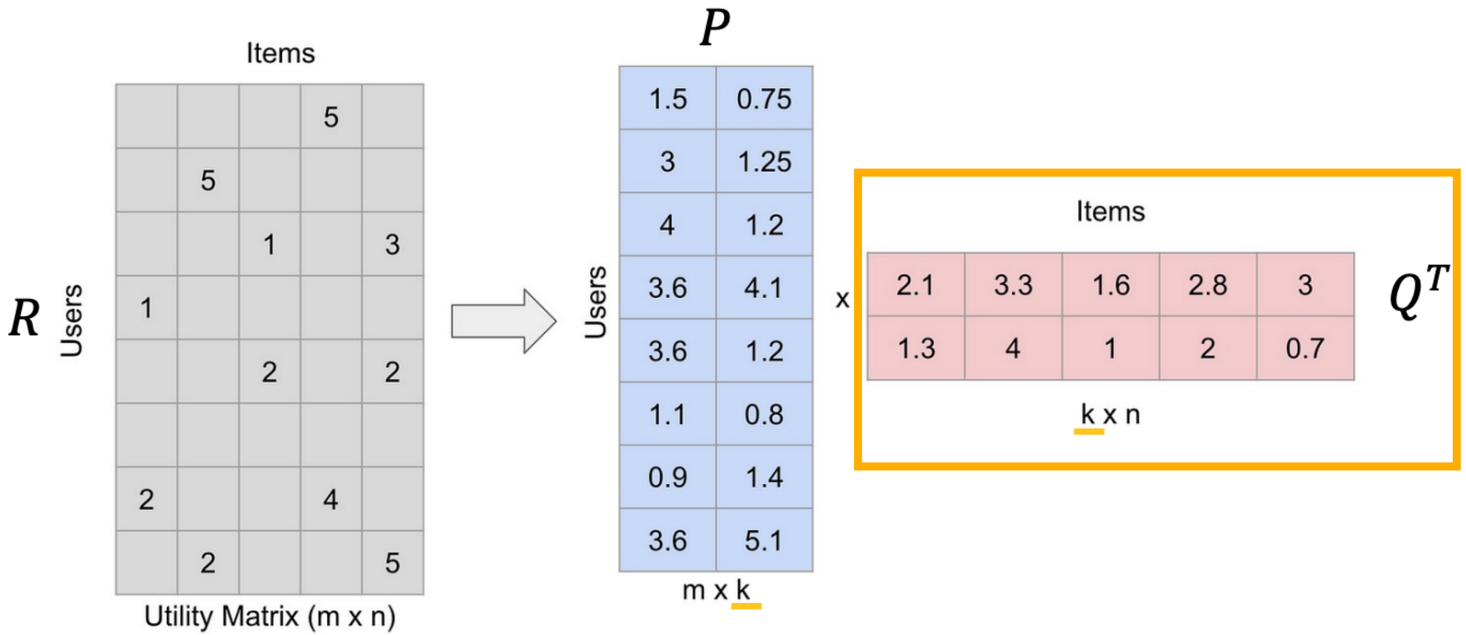
$$L = \sum_{(i,j)} \left(r_{ij} - \bar{p}_i \cdot \bar{q}_j^T \right)^2 + \alpha \left(\|\bar{p}_i\|^2 + \|\bar{q}_j\|^2 \right)$$

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

Почему так можно делать

Второй подход основан на особенностях функционала (1.1) и называется ALS (alternating least squares). Можно показать, что этот функционал не является выпуклым в совокупности по P и Q, но при это становится выпуклым, если зафиксировать либо P, либо Q. Более того, оптимальное значение P при фиксированном Q (и наоборот) можно выписать аналитически, — но оно будет содержать обращение матрицы:

$$p_u = \left(\sum_{i:\exists r_{ui}} q_i q_i^T \right)^{-1} \sum_{i:\exists r_{ui}} r_{ui} q_i;$$

$$q_i = \left(\sum_{u:\exists r_{ui}} p_u p_u^T \right)^{-1} \sum_{u:\exists r_{ui}} r_{ui} p_u;$$

Alternating Least Squares

Итоговый процесс оптимизации функции потерь будет иметь следующий вид.

В цикле до сходимости:

- Фиксируем матрицу X (скрытые представления пользователей);
- Решаем задачу L2-регуляризованной регрессии для каждого товара и находим оптимальную матрицу Y;
- Фиксируем матрицу Y (скрытые представления объектов);
- Решаем задачу L2-регуляризованной регрессии для каждого пользователя и находим оптимальную матрицу X;

Implicit Alternating Least Squares

- ALS предполагает явный фидбек, а его обычно мало
- Хотим научиться работать с неявным (implicit) фидбеком

Implicit Alternating Least Squares

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

x_u — эмбеддинг пользователя

y_i — эмбеддиннг 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}

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

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

Обобщения ALS и iALS

- Обе модели: и ALS, и Imlicit ALS – можно несколько усложнить, вместо $r_{ui} \approx x_u y_i$ рассмотрев $r_{ui} \approx x_u y_i + b_u + b_i + \mu$. В таком случае b_i и b_j играют роль некоторых априорных усреднённых оценок пользователя и объекта соответственно, а μ является глобальной априорной константой.
- В модели IALS мы обычно полагаем элементы p_{ui} равными 1 во всех случаях, когда имело место взаимодействие, но можем использовать и другие значения, в том числе зависящие от того, что ещё нам известно о пользователях и объектах.
- Для уверенности $c_{uv} = 1 + \alpha \|r_{ui}\|$ для IALS необязательно использовать 1 в качестве значения по умолчанию. Например, события «пользователь не посмотрел популярный фильм» и «пользователь не посмотрел редкий фильм» могут иметь для нас разный вес.

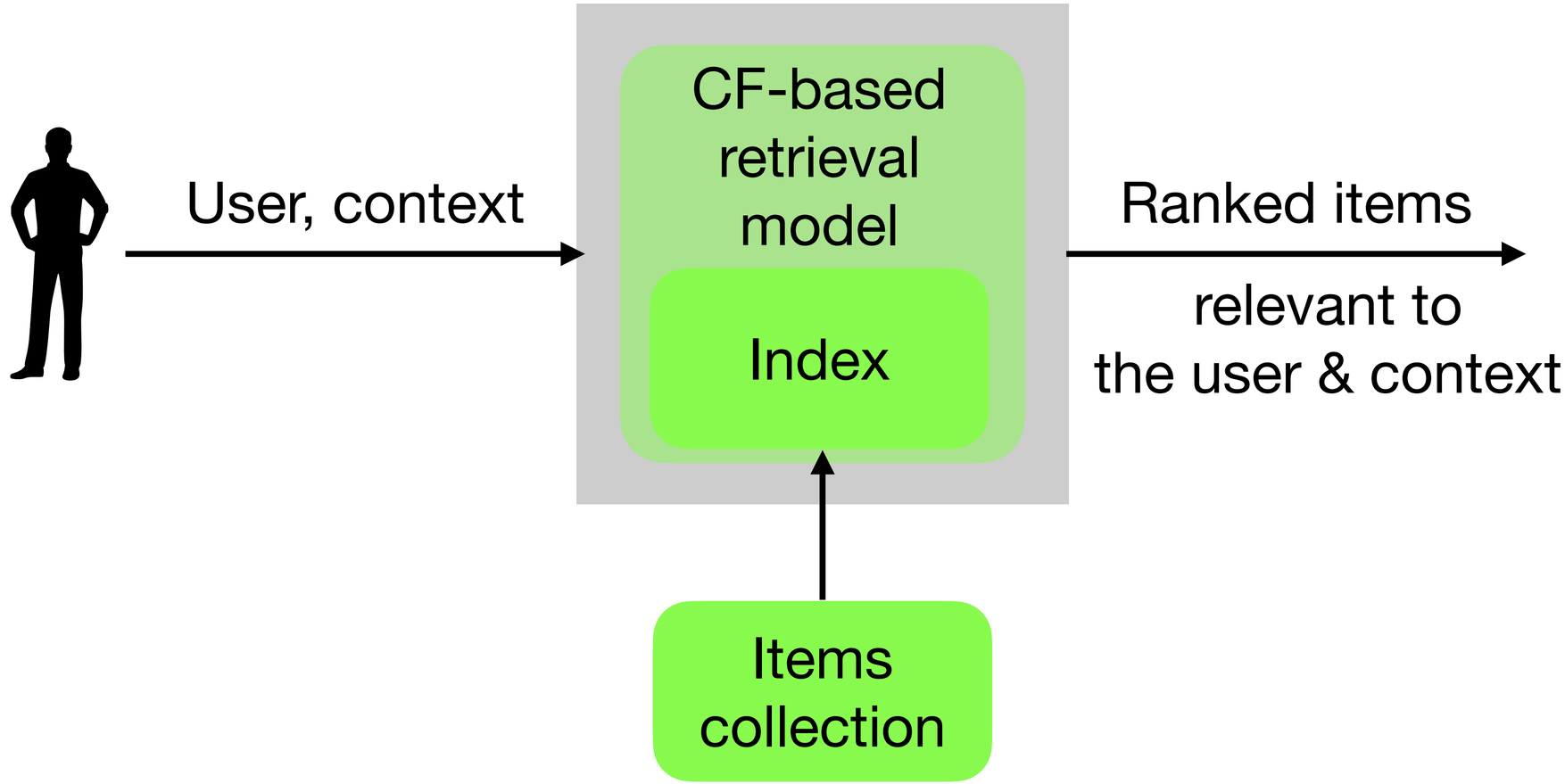
Выводы

- **Коллаборативная фильтрация** (Collaborative Filtering) — это набор методов для построения рекомендательных систем

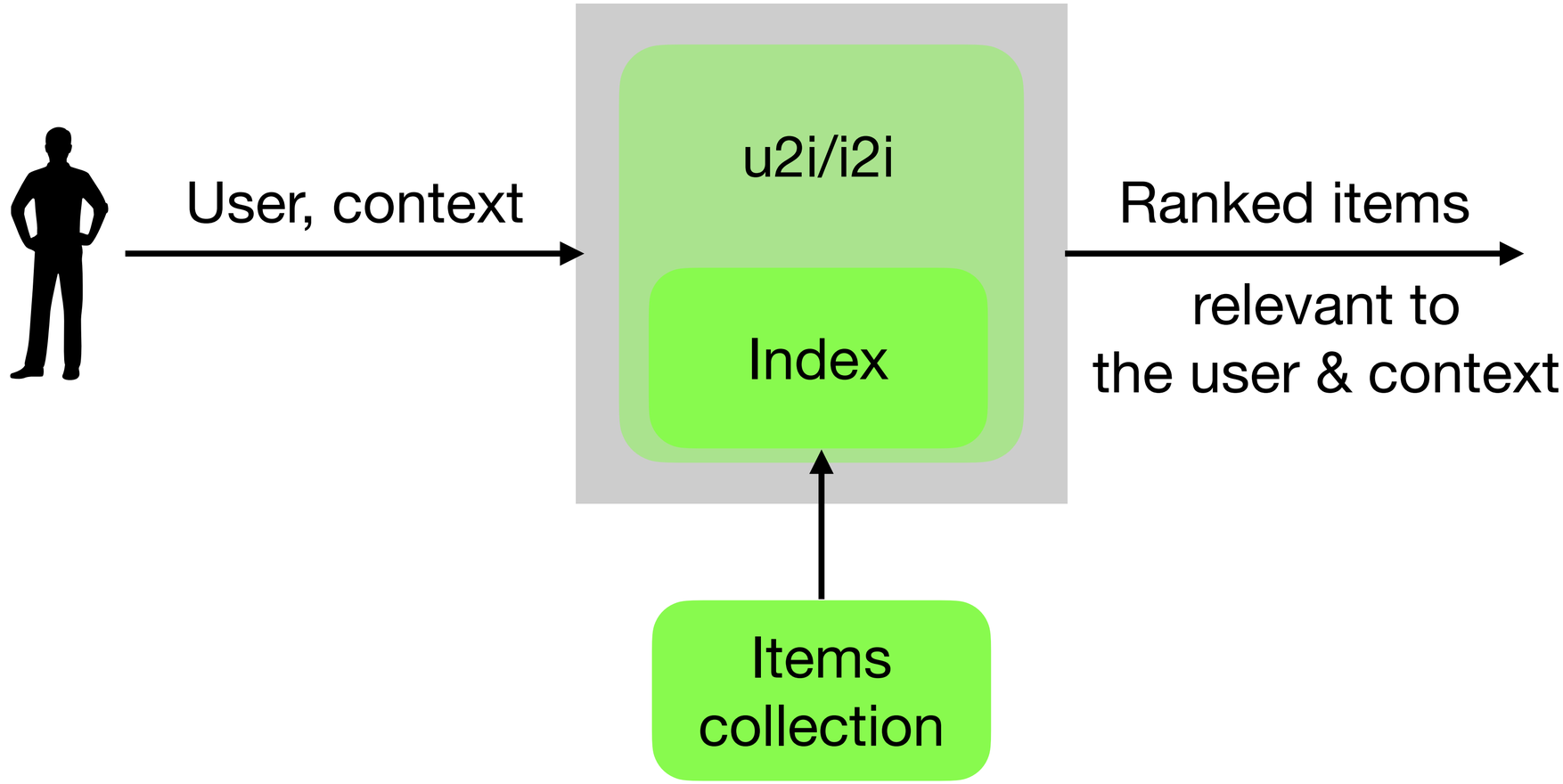
- **Корреляционные модели** — простые, но устаревшие.

- **Латентные модели** на основе матричных разложений обладают рядом преимуществ:
  - сокращается объём хранимых данных
  - привлекаются внешние данные для «холодного старта»
  - неотрицательные и вероятностные эмбединги интерпретируются как векторы «интересов» или «тем»
  - легко добавляются регуляризаторы для многокритериальной оптимизации качества рекомендаций

Рекомендательная система



Рекомендательная система



Вопросы