

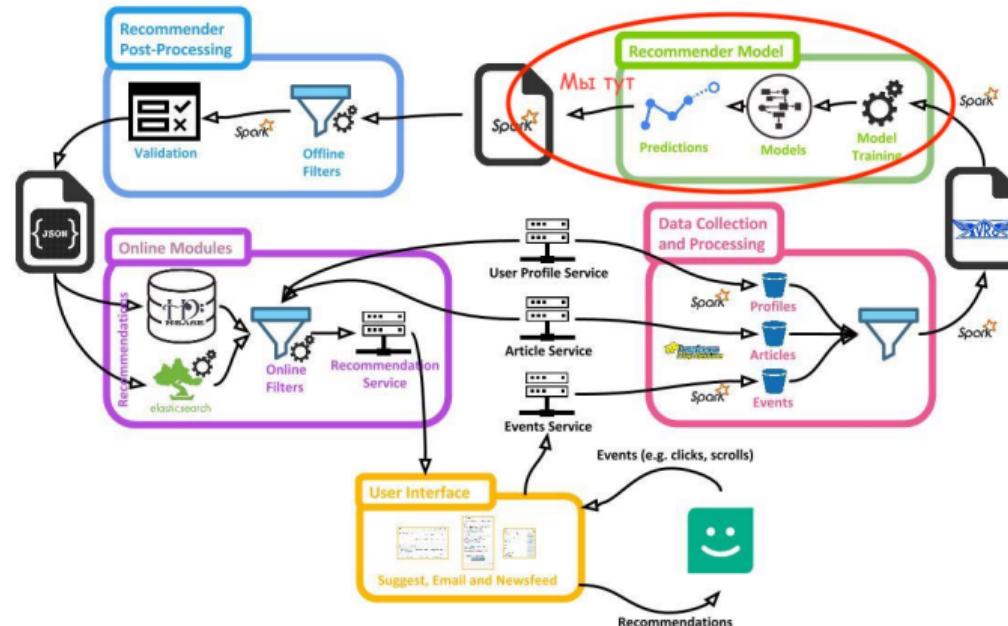
Классические алгоритмы рекомендаций

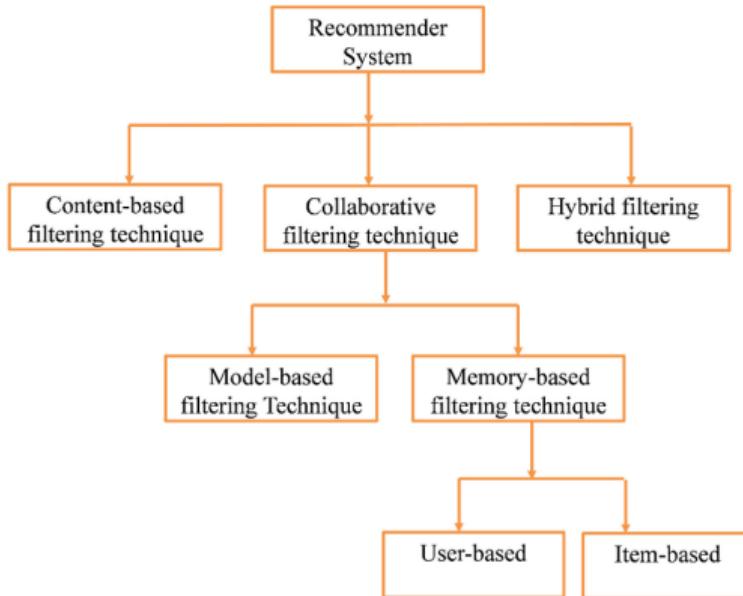
Николай Анохин

9 октября 2023 г.



Контекст





Model-based Collaborative Filtering



Model-based CF

Идея

Выучим модель, которая поможет заполнить “пробелы” в user-item матрице.

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	4			5	1		
B	5	5	4	?	?	2	
C				2	4	5	
D		3					3



Бейзлайн [KBV09]

Модель

$$b_{ui} = \mu + b_u + b_i$$

- μ – средний рейтинг
- b_u – bias пользователя
- b_i – bias айтема

Оптимизируем

$$\sum_{u,i} (b_{ui} - \mu - b_u - b_i)^2 + \lambda_1 b_u^2 + \lambda_2 b_i^2 \rightarrow \min_{b_u, b_i}$$



SVD

Модель

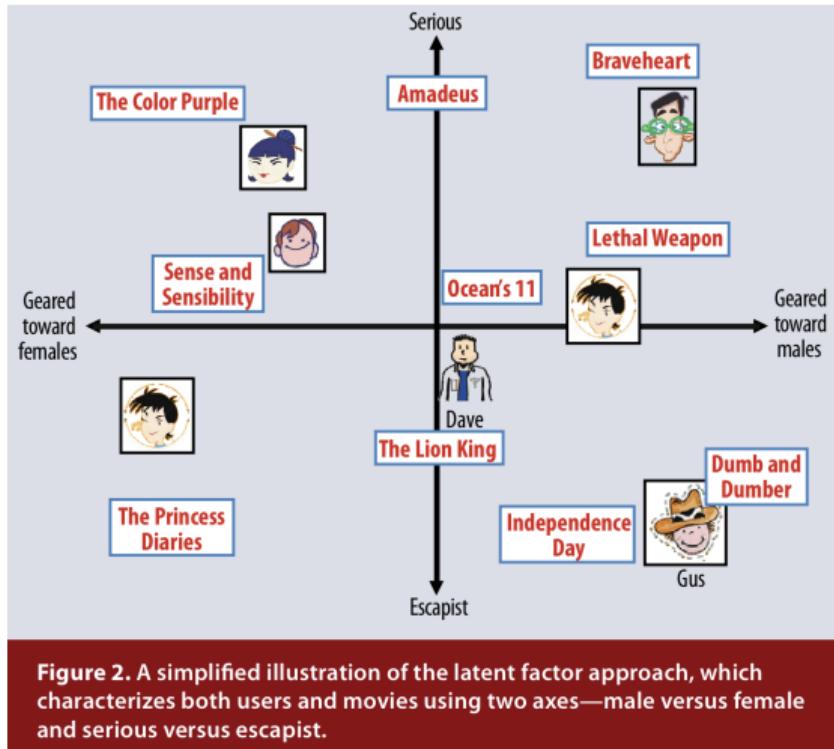
$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T p_u$$

- q_i – латентное представление айтема
- p_u – латентное представление пользователя

Оптимизируем

$$\sum_{u,i} (r_{ui} - \mu - b_u - b_i - q_i^T p_u)^2 + \lambda(b_u^2 + b_i^2 + \|q_i\|^2 + \|p_u\|^2) \rightarrow \min_{b_u, b_i, p_u, q_i}$$





Как оптимизировать

$$\sum_{u,i} (r_{ui} - \mu - b_u - b_i - q_i^T p_u)^2 + \lambda(b_u^2 + b_i^2 + \|q_i\|^2 + \|p_u\|^2)$$

- ALS [HKV08]
 1. фиксируем p_u, b_u , оптимизируем q_i, b_i – получем линрег 1
 2. фиксируем q_i, b_i , оптимизируем p_u, b_u – получем линрег 2
 3. повторяем до сходимости
- SGD



SVD++

Модель

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T \left(p_u + \frac{1}{\sqrt{|R(u)|}} \sum_j y_j \right)$$

- y_i – латентное представление айтемов, на которые пользователь дал implicit feedback до оценки айтема i

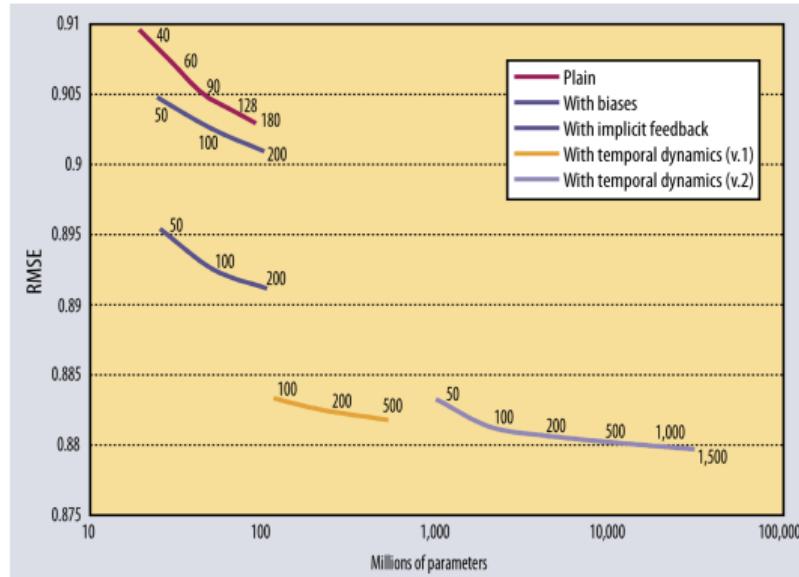
Time SVD++

Модель

$$\hat{r}_{ui} = \mu + b_u(t_{ui}) + b_i(t_{ui}) + q_i^T \left(p_u(t_{ui}) + \frac{1}{\sqrt{|R(u)|}} \sum_j y_j \right)$$

- t_{ui} – время, когда пользователь оценил айтем

Netflix Prize



September 21, 2009, the grand prize of US\$1,000,000 was given to the BellKor's Pragmatic Chaos team which bested Netflix's own algorithm for predicting ratings by 10.06



LightFM [Kul15]

Модель

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T p_u$$

$$q_i = \sum_{j \in f_i} v_j, \quad p_u = \sum_{j \in f_u} w_j$$

- f_i – признаки айтема
- v_j – латентное представление признаков айтема
- f_u – признаки пользователя
- w_j – латентное представление признаков пользователя



Альтернативные loss-функции: classification vs regression

Случай implicit feedback похож скорее на задачу классификации, чем регрессии

$$\hat{p}_{ui}(r = 1) = \sigma(\mu + b_u + b_i + q_i^T p_u)$$

Лосс: кросс-энтропия



Альтернативные loss-функции: ranking with BPR [RFGST09]

Правильное ранжирование важнее, чем точное предсказание рейтинга / фидбэка

$$\hat{p}(u \text{ prefers } i \text{ to } j) = \sigma(\hat{x}_{uij}) = \sigma(\hat{x}_{ui} - \hat{x}_{uj})$$

Лосс:

$$-\sum \log p(u \text{ prefers } i \text{ to } j) + \lambda \|\theta\|^2$$



Альтернативные loss-функции: WARP [Wil16]

Можно умно семплировать негативные примеры – так, чтобы сложность негативных примеров увеличивалась, когда модель становится точнее.

Дано: пользователь + позитивный айтем

1. Семплируем негативные, пока не найдем неправильно отранжированную пару.
Делаем шаг обновления.
2. Чем больше пришлось семплировать, тем меньше learning rate шага обновления.



Реализации

	Spark	LightFM	Implicit
SGD		✓	
ALS	✓		✓
RMSE	✓		✓
logistic		✓	✓
BPR		✓	✓
WARP		✓	
GPU			✓



Простые автоэнкодеры

SLIM [NK11] / EASE [Ste19]

$$\hat{A} = AW,$$

где

- A – матрица интеракций
- \hat{A} – предсказанные интеракции
- $W_{:,j}$ – колонка весов агрегации для j айтема ($W_{ij} \geq 0$ (SLIM), $W_{ij} = 0$).

Плюсы

- Качество рекомендаций: новизна как у neighbourhood-based CF, но можно заполнить все пробелы в user-item матрице
- Большой выбор моделей и лоссов
- Можно переформулировать как нейронную сеть

Минусы

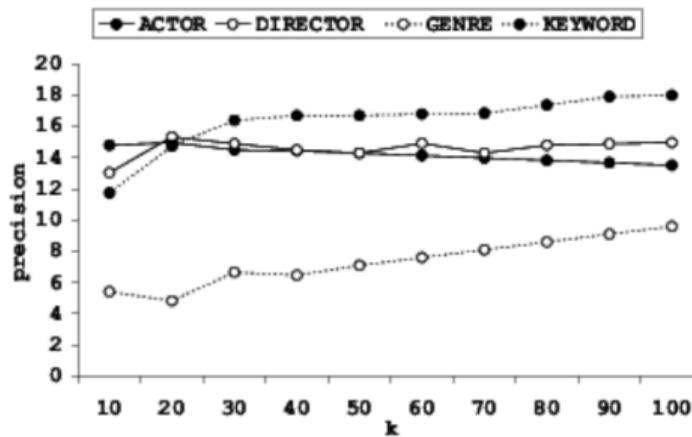
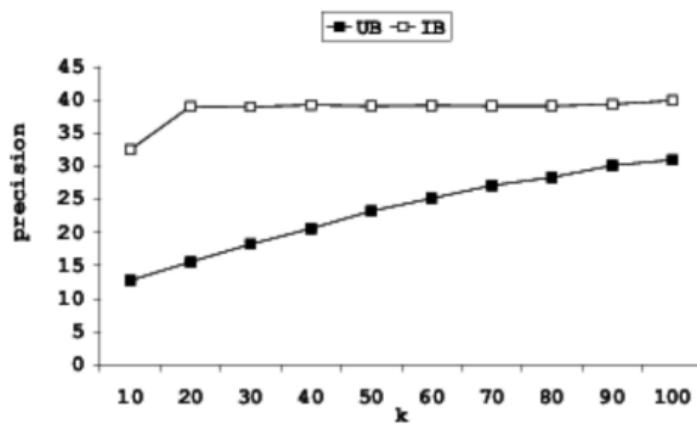
- Сложные алгоритмы оптимизации
- Холодный старт как пользователей, так и айтемов нужно отдельно решать



Итоги



CB vs CF



CF Flavors

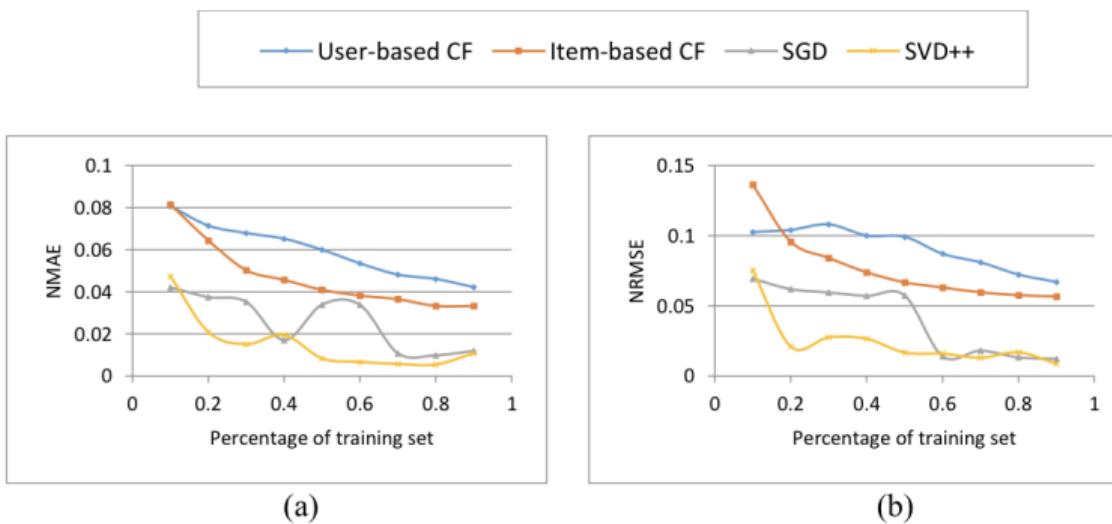


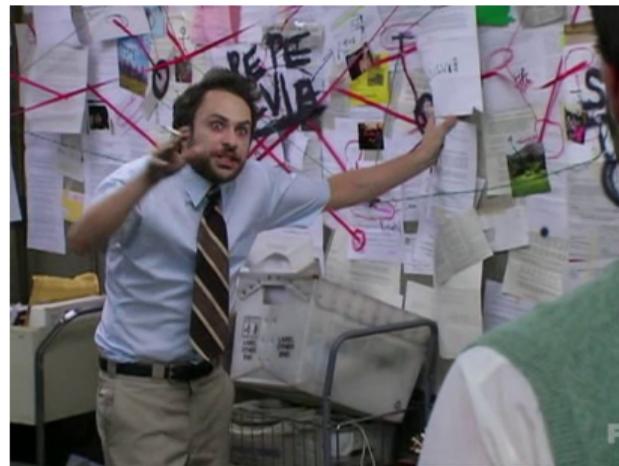
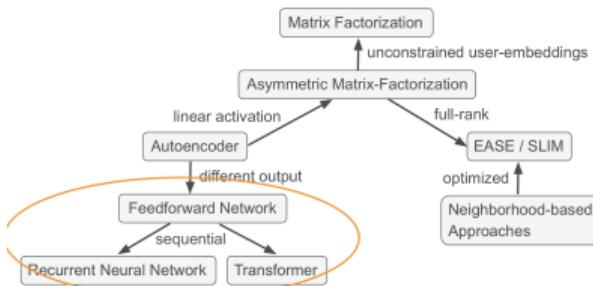
Figure 6. Performance comparison of different recommendation algorithms at different training ratios

Реальные рекомендеры почти всегда hybrid: применяются как CF, так и СВ подходы

“Дефолтным” алгоритмом рекомендаций считается матричная факторизация



В следующий раз



Литература I

-  Yifan Hu, Yehuda Koren, and Chris Volinsky, *Collaborative filtering for implicit feedback datasets*, Proceedings of the 2008 Eighth IEEE International Conference on Data Mining (USA), ICDM '08, IEEE Computer Society, 2008, p. 263–272.
-  Yehuda Koren, Robert Bell, and Chris Volinsky, *Matrix factorization techniques for recommender systems*, Computer **42** (2009), no. 8, 30–37.
-  Maciej Kula, *Metadata embeddings for user and item cold-start recommendations*, Proceedings of the 2nd Workshop on New Trends on Content-Based Recommender Systems co-located with 9th ACM Conference on Recommender Systems (RecSys 2015), Vienna, Austria, September 16-20, 2015. (Toine Bogers and Marijn Koolen, eds.), CEUR Workshop Proceedings, vol. 1448, CEUR-WS.org, 2015, pp. 14–21.
-  Xia Ning and George Karypis, *Slim: Sparse linear methods for top-n recommender systems.*, ICDM (Diane J. Cook, Jian Pei, Wei Wang, Osmar R. Zaïane, and Xindong Wu, eds.), IEEE Computer Society, 2011, pp. 497–506.



Литература II

-  Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme, *Bpr: Bayesian personalized ranking from implicit feedback*, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence (Arlington, Virginia, USA), UAI '09, AUAI Press, 2009, p. 452–461.
-  Harald Steck, *Embarrassingly shallow autoencoders for sparse data*, The World Wide Web Conference (New York, NY, USA), WWW '19, Association for Computing Machinery, 2019, p. 3251–3257.
-  Benjamin Wilson, *Warp loss for implicit-feedback recommendation*, Mar 2016.

