Recommender systems

Matrix Factorization models

Lecture 3
Fall 2023

Anna Volodkevich

Содержание

Задача рекомендаций как задача восстановления пропущенных значений в матрице взаимодействий

SVD, Truncated SVD, Funk SVD

ALS, Implicit ALS

Recap

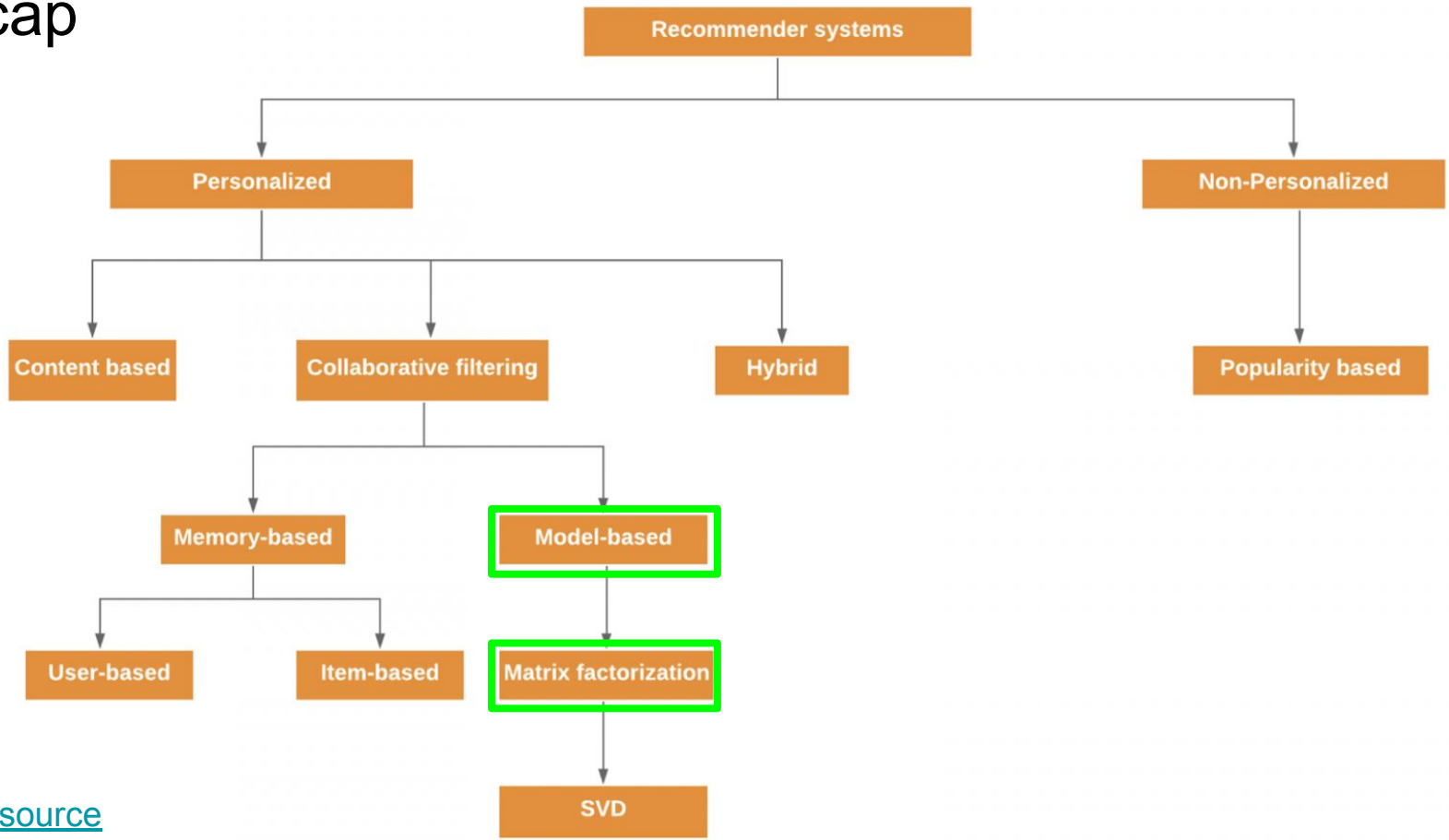


Image source

Recap

Model-based methods

In model-based methods, machine learning and data mining methods are used in the context of predictive models. In cases where the model is parameterized, the parameters of this model are learned within the context of an optimization framework. Some examples of such model-based methods include decision trees, rule-based models, Bayesian methods and **latent factor models**. Many of these methods, such as latent factor models, have a high level of coverage even for sparse ratings matrices.

Latent factor models, such as matrix factorization, transform both items and users to the same latent factor space. The latent space tries to explain ratings by characterizing both products and users on factors automatically inferred from user feedback.

Задача рекомендаций как задача восстановления пропущенных значений

- Взаимодействия user-ов и item-ов можно представить в матричнов виде
- Возможно, у user-ов и item-ов есть скрытые признаки
- Подход к решению задачи будет отличаться в зависимости от типа feedback

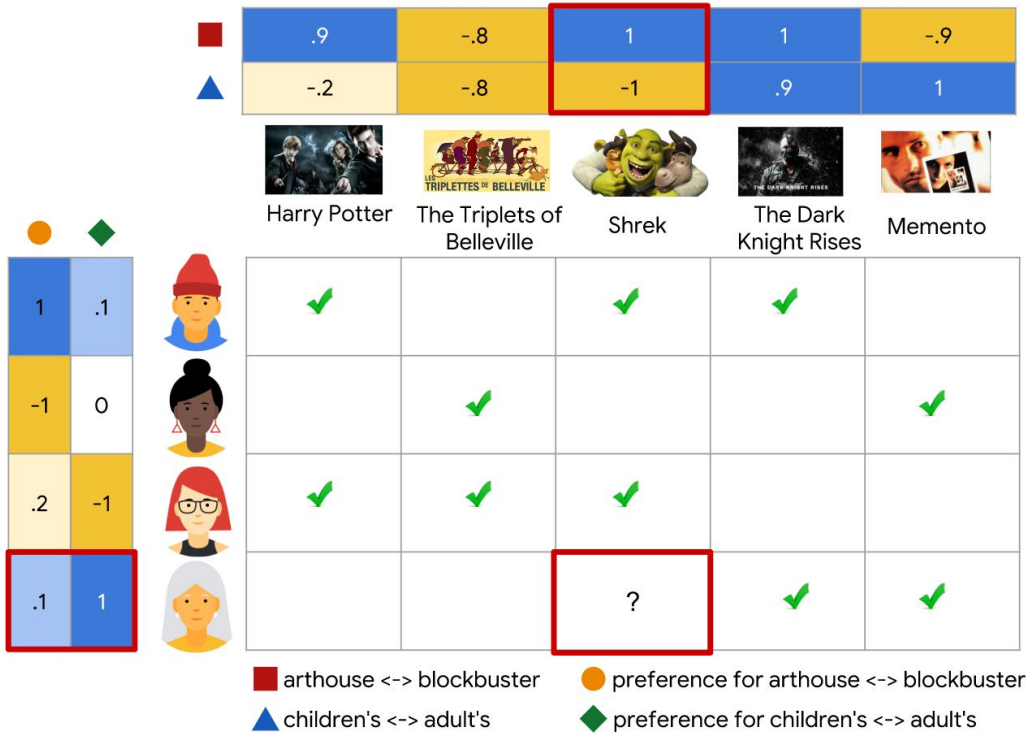
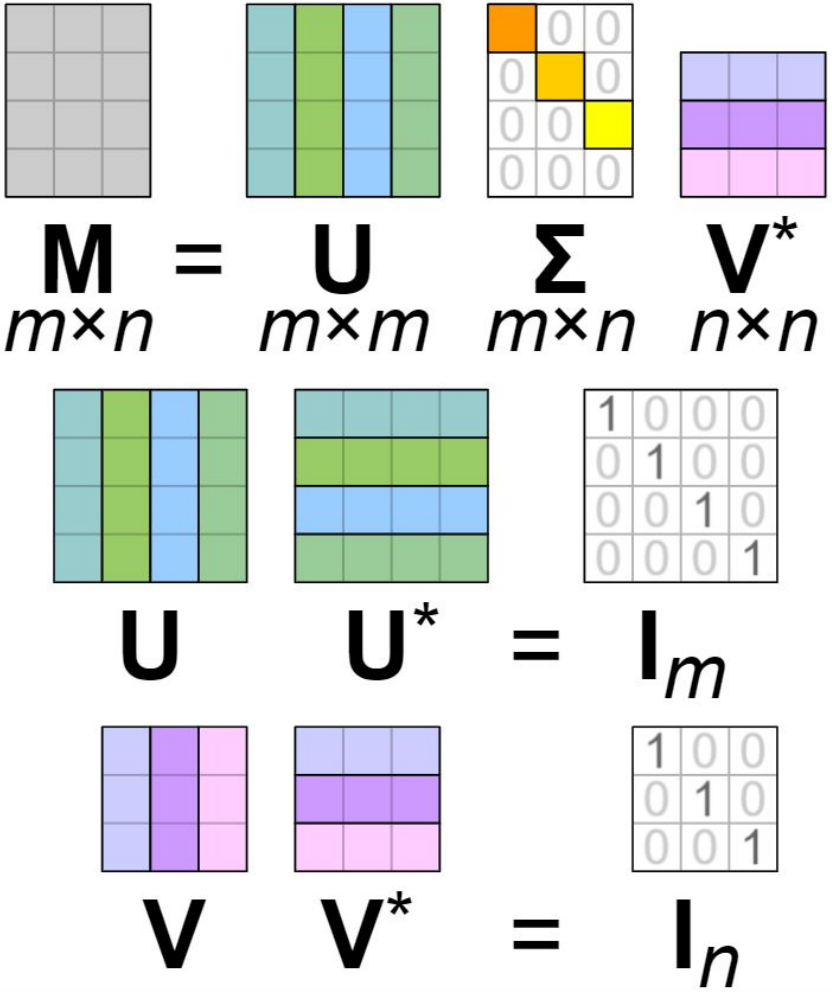


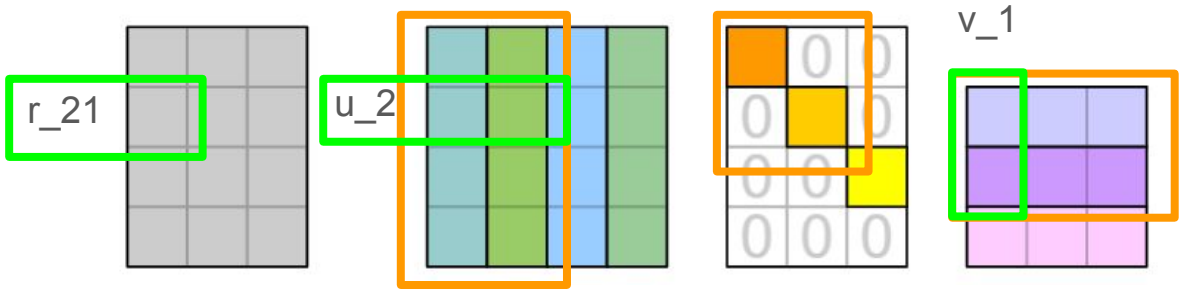
Image source

SVD: повторение

- Σ - диагональная m×n-матрица с диагональю из невозрастающих сингулярных чисел $\sigma 1, \ldots, \sigma n$,
- U и V – ортогональные матрицы.



Truncated SVD



k << n, k << m

$$R \approx U_k \Sigma_k V_k^T \qquad ||R - U_k \Sigma_k V_k^t||_F^2 \to \min$$

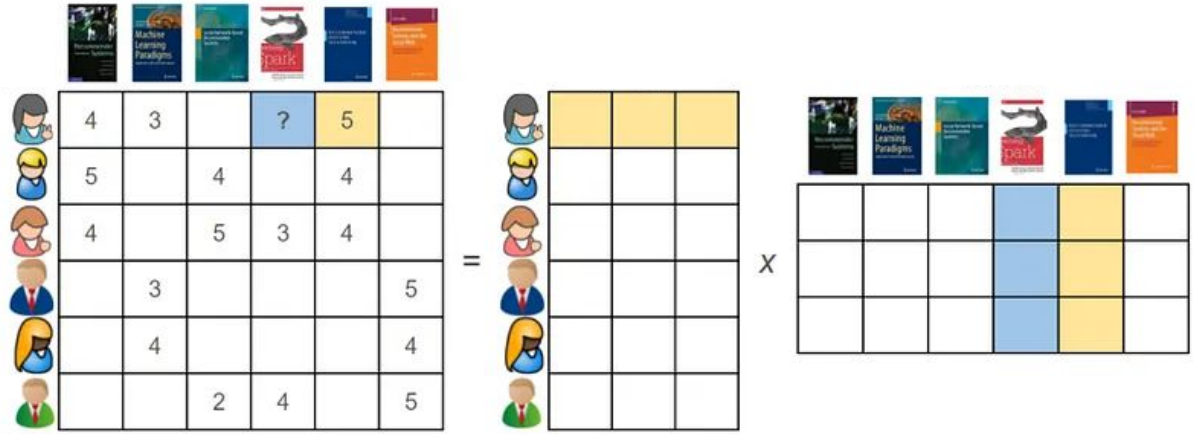
$$r_{ij} = \sum_{m=1}^{k} \sigma_m u_{im} v_{jm}$$

- Интерпретация: совпадение “тем” пользователя и айтема с учетом важности влияния “темы” на рейтинг
- Наилучшее приближение ранга k по норме Фробениуса

Визуализация

Funk SVD

$$R \approx PQ^T$$



$$\sum_{(i,j)\in R} (r_{ij} - p_i^T q_j)^2 + \sum_{i=1}^{m} \lambda_1 ||p_i||^2 + \sum_{j=1}^{n} \lambda_2 ||q_j||^2 \to \min$$

Image source

Повторение: cosine similarity, dot product

Dot product

$$\mathbf{a} \cdot \mathbf{b} = \|\mathbf{a}\| \, \|\mathbf{b}\| \cos\theta,$$

where θ is the angle between $\mathbf{a}$ and $\mathbf{b}$.

Cosine similarity

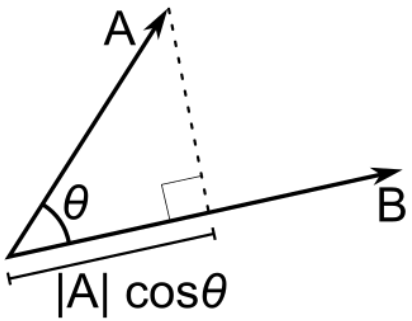
$$S_C(A, B) := \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|}$$

Проекция вектора a на b:

$$a_b = \|\mathbf{a}\| \cos\theta,$$

.

$$\mathbf{a} \cdot \mathbf{b} = a_b \, \|\mathbf{b}\| = b_a \, \|\mathbf{a}\| \, .$$



Визуализация поиска ближайших по косинусу

- получили решение и для персональных рекомендаций и для item2item
- можно использовать методы поиска ближайших соседей для быстрого нахождения top-k рекомендаций

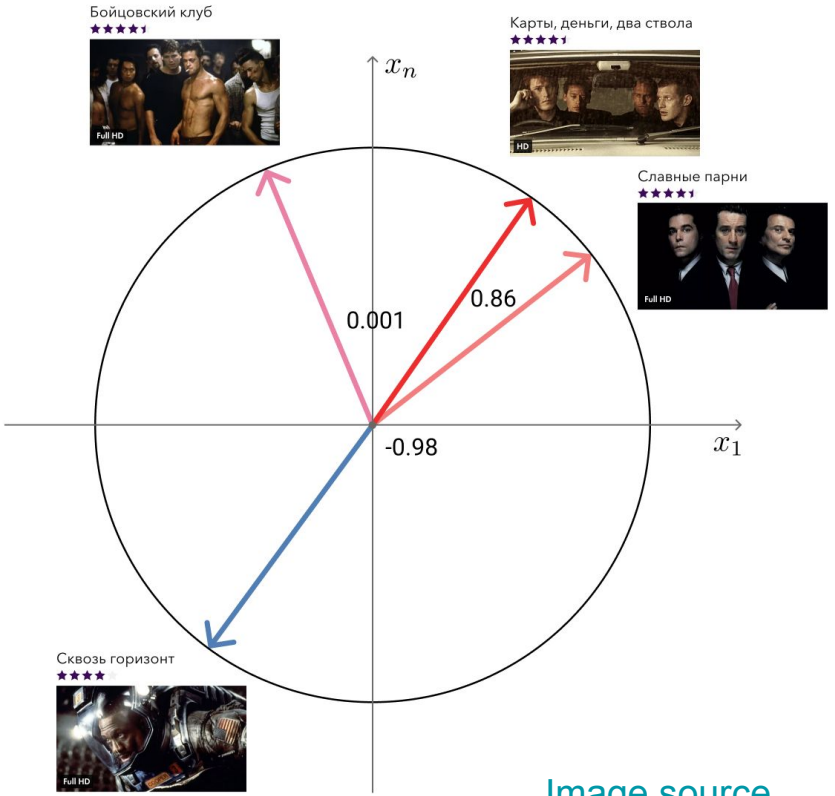


Image source

SVD: плюсы и минусы

Минусы:

- используем только взаимодействия (не учитываем признаки и контекст и последовательность взаимодействий)
- Truncated SVD: считаем, что на пустых местах матрицы находятся нули
- Funk SVD: не учитываем информацию об отсутствии взаимодействий
- перекос в сторону популярных: если у объекта много оценок, оценки для него в восстановленной матрице будут выше

Плюсы:

- сильный baseline “из коробки”
- полученные вектора можно использовать для быстрого inference с помощью методов приближенного поиска соседей (ANN)
- Funk SVD можно адаптировать под свои задачи и данные, изменяя функцию потерь (примеры дальше)

SVD: вариации и усложнения

- Учитываем bias-ы в оценках

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

$$\min_{b_*, q_*, p_*} \sum_{(u,i)\in\mathcal{K}} (r_{ui} - \mu - b_i - b_u - q_i^T p_u)^2 + \lambda_4 (b_i^2 + b_u^2 + \|q_i\|^2 + \|p_u\|^2) \,.$$

- SVD++: учитываем implicit feedback

$$\hat{r}_{ui} = \mu + b_i + b_u + q_i^T \left(p_u + |\mathrm{R}(u)|^{-\frac{1}{2}} \sum_{j\in\mathrm{R}(u)} y_j \right)$$

y_j - вектора айтемов, обученные по implicit feedback, можно учесть несколько типов implicit feedback

- timeSVD, timeSVD++: попытка учесть изменения предпочтений пользователей во времени
- …

Recommender Systems Handbook, Second Edition, Springer, 2011

Alternating least squares

$$\sum_{(i,j)\in R} (r_{ij} - p_i^T q_j)^2 + \sum_{i=1}^{m} \lambda_1 ||p_i||^2 + \sum_{j=1}^{n} \lambda_2 ||q_j||^2 \to \min$$

- Идея: оптимизировать P и Q попеременно, а не одновременно
- Шаги: инициализируем случайно P и Q
- В цикле:
  - Фиксируем P, находим решение для Q
  - Фиксируем Q, находим решение для P
- Решения для отдельных items/users можно искать независимо

Alternating least squares

Algorithm 1 ALS for Matrix Completion

Initialize X, Y

repeat

for $u = 1 \dots n$ **do**

$$x_u = \left(\sum_{r_{ui} \in r_{u*}} y_i y_i^\intercal + \lambda I_k \right)^{-1} \sum_{r_{ui} \in r_{u*}} r_{ui} y_i \tag{2}$$

end for

for $i = 1 \dots m$ **do**

$$y_i = \left(\sum_{r_{ui} \in r_{*i}} x_u x_u^\intercal + \lambda I_k \right)^{-1} \sum_{r_{ui} \in r_{*i}} r_{ui} x_u \tag{3}$$

end for

until convergence

source

Alternating least squares for implicit feedback (iALS)

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

p_{ui} - факт наличия фидбека

c_{ui} - степень уверенности

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

$f \times f$ matrix $Y^T Y$ in time $O(f^2 n)$. For each user u, let us define the diagonal $n \times n$ matrix C^u where $C^u_{ii} = c_{ui}$, and also the vector $p(u) \in \mathbb{R}^n$ that contains all the preferences by u (the p_{ui} values). By differentiation we find an analytic expression for x_u that minimizes the cost function (3):

$$x_u = (Y^T C^u Y + \lambda I)^{-1} Y^T C^u p(u) \quad \blacktriangleright \quad x_u = (Y^T Y + \lambda I + \sum_{\forall i | p_{ui} \neq 0} (c_{ui} - 1) y_i y_i^T)^{-1} (\sum_{\forall i | p_{ui} \neq 0} c_{ui} p_{ui} y_i)$$

source

ALS/iALS: плюсы и минусы

Минусы:

- используем только взаимодействия (не учитываем признаки и контекст(не учитываем признаки, контекст и последовательность взаимодействий)
- только квадратичная функция потерь

Плюсы:

- сильный baseline “из коробки”
- подходит для любого типа фидбека
- хорошо параллелится
- существуют хорошие готовые реализации для работы в локальном и кластерных режимах (implicit, spark mllib)
- возможность обновить вектора пользователей при появлении новых взаимодействий без перезапуска модели
- полученные вектора можно использовать для быстрого inference с помощью методов приближенного поиска соседей (ANN)

Другие модели латентных факторов

- NMF (Non-Negative Matrix Factorization)
- LDE (Latent Dirichlet allocation)
- PLSA (Probabilistic Latent Semantic Analysis)

Дополнительные материалы

Статьи

Habr: Рекомендательные системы: идеи, подходы, задачи

Habr: Как работают рекомендательные системы. Лекция в Яндексе

Лекция по рекомендательным системам из курса ВШЭ по машинному обучению

Книги:

“Recommender Systems. The Textbook”, 2016, Springer

“Recommender Systems Handbook”, 2011, Springer

Библиотеки

implicit: популярная реализация iALS и item-based KNN