

Рекомендательные системы

Методы матричной факторизации



Проверить, идет ли запись

Меня хорошо видно && слышно?



Тема вебинара

Методы матричной факторизации



Вероника Иванова

DS RecSys в Sber AI Lab, до этого в Домклик

провожу исследования в области
sequential recommender systems

руководитель курса RecSys
преподаватель курсов ML, ML Adv, Data Analyst в ОТУС

Контакты: @nica_vera



Маршрут вебинара

Методы матричной факторизации

Pure SVD

Funk SVD

ALS, iALS

Практика

Цели вебинара

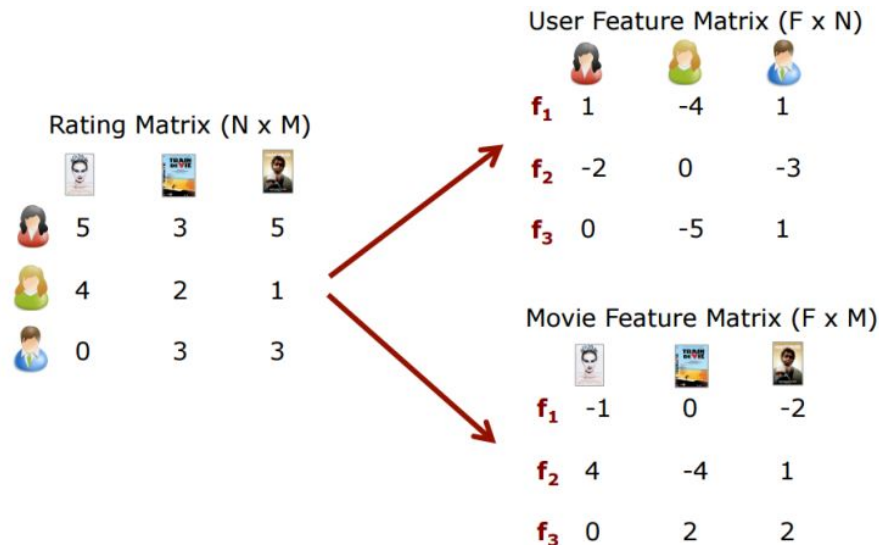
К концу занятия вы сможете

1. Рассмотреть метод матричной факторизации
2. Узнать, как использовать SVD разложение в задаче рекомендации
3. Познакомиться с другими популярными моделями матричной факторизации

Методы матричной факторизации

Матричная факторизация - это класс алгоритмов коллаборативной фильтрации, которые работают путем разложения матрицы взаимодействия пользователь - объект на произведение двух (или трех) прямоугольных матриц меньшей размерности.

Полученные при разложении матрицы будут характеризовать интересы пользователей и свойства объектов.



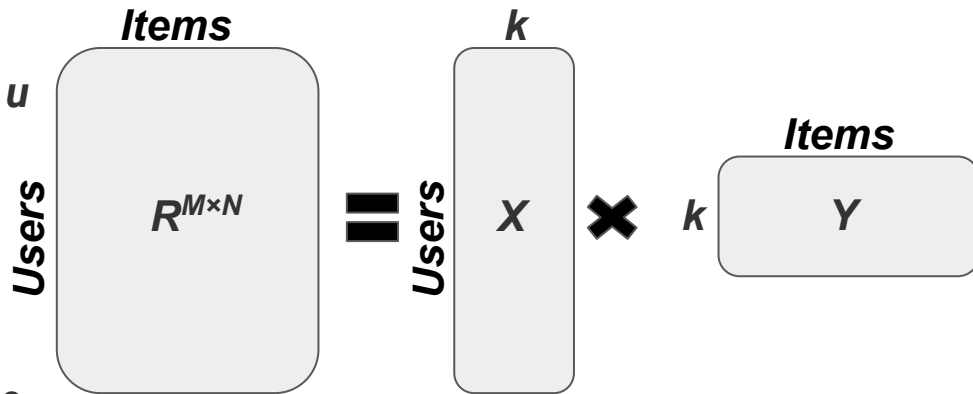
Получение предсказаний

Для предсказания оценки пользователя u для объекта i необходимо вычислить скалярное произведение вектора пользователя x_u на вектор объекта y_i .

$$r_{ui} = \langle x_u, y_i \rangle$$

Для предсказания всех оценок в матрице взаимодействий нужно вычислить произведение матриц пользователей X и объектов Y .

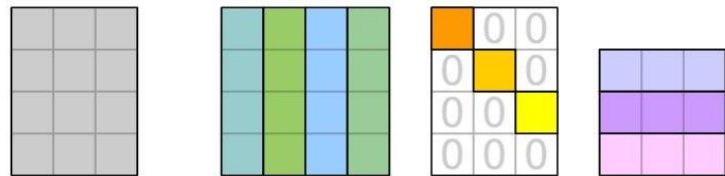
$$R^{M \times N} = X^{M \times k} \cdot Y^{k \times N}$$



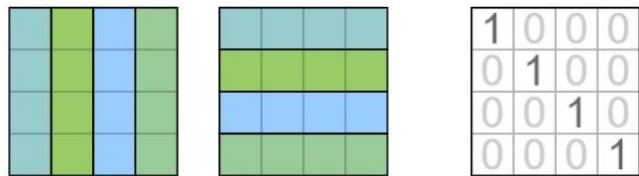
Pure SVD

Σ - матрица размера $m \times n$ с неотрицательными элементами, у которой элементы, лежащие на главной диагонали - это сингулярные числа, а все элементы, не лежащие на главной диагонали, равны нулю.

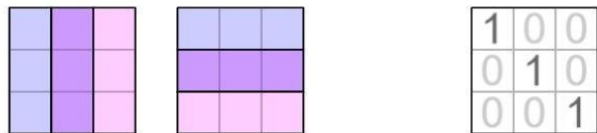
U и V - ортогональные матрицы размера $m \times m$ и $n \times n$, состоящие из левых и правых сингулярных векторов.



$$\begin{matrix} \mathbf{M} \\ m \times n \end{matrix} = \begin{matrix} \mathbf{U} \\ m \times m \end{matrix} * \begin{matrix} \mathbf{\Sigma} \\ m \times n \end{matrix} * \begin{matrix} \mathbf{V}^T \\ n \times n \end{matrix}$$



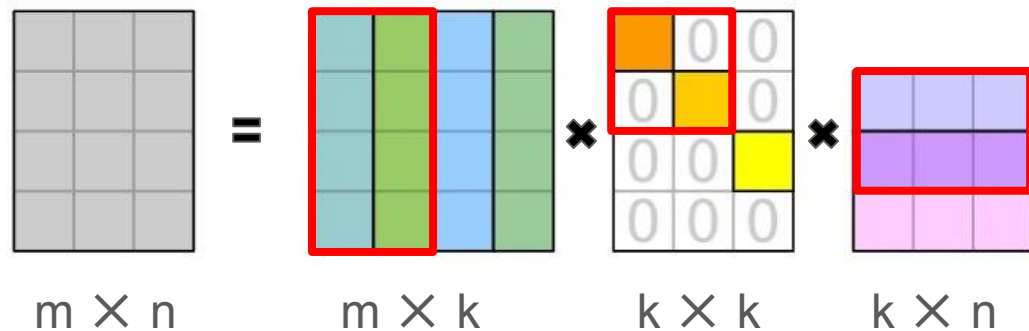
$$\mathbf{U} * \mathbf{U}^T = \mathbf{I}_m$$



$$\mathbf{V} * \mathbf{V}^T = \mathbf{I}_n$$

Truncated SVD

- Получение рейтингов может быть интерпретировано как вычисление меры сходства между пользователями и объектами в пространстве латентных признаков.
- Можно получать рейтинги, перемножая исходную матрицу взаимодействий на квадрат матрицы объектов.
- Наилучшее приближение исходной матрицы с точки зрения нормы Фробениуса.



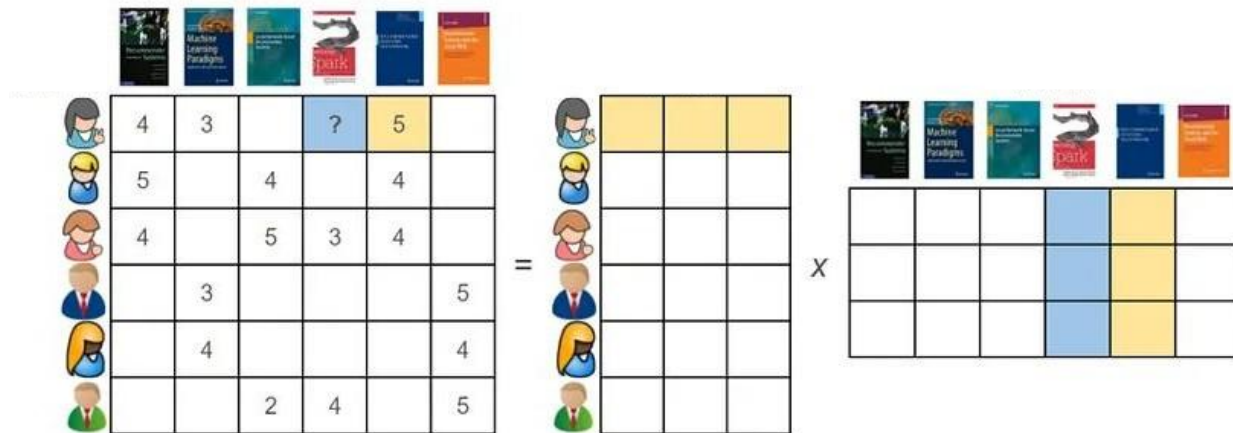
$$k \ll n, k \ll m$$

$$R \approx U_k \Sigma_k V_k^T$$
$$R \approx M V_k V_k^T$$
$$\|R - U_k \Sigma_k V_k^T\|_F^2 \rightarrow \min$$

Funk SVD

Представляем матрицу рейтингов как произведение матрицы пользователей на матрицу объектов.

Обучение происходит при помощи градиентного спуска.



$$R \approx XY^T$$

$$\sum_{(u,i) \in R} (r_{ui} - \langle x_u, y_i \rangle)^2 + \lambda (||x_u||^2 + ||y_i||^2) \rightarrow \min$$

Funk SVD

При обучении модели можно учитывать смещение оценок:

- μ - глобальное среднее всех рейтингов
- b_i - смещение оценок объектов
- b_u - смещение оценок пользователя

$$\hat{r}_{ui} = \mu + b_i + b_u + \langle x_u, y_i \rangle$$

$$\sum_{(u,i) \in R} (r_{ui} - \hat{r}_{ui})^2 + \lambda (b_i^2 + b_u^2 + \|x_u\|^2 + \|y_i\|^2) \rightarrow \min$$

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

$$b_u = b_u + \alpha \cdot (err - \lambda b_u)$$

$$b_i = b_i + \alpha \cdot (err - \lambda b_i)$$

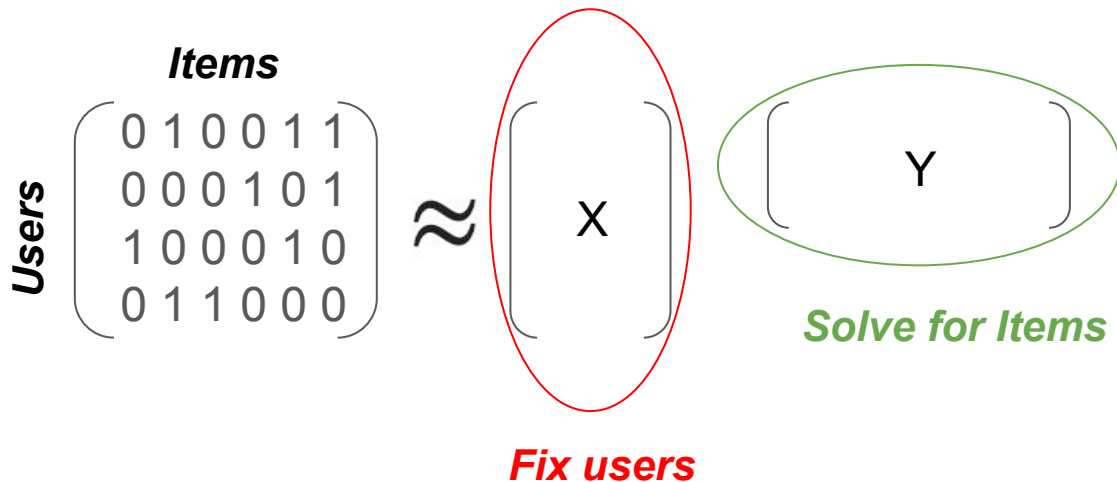
$$x_u = x_u + \alpha \cdot (err \cdot y_i - \lambda x_u)$$

$$y_i = y_i + \alpha \cdot (err \cdot x_u - \lambda y_i)$$

где λ - коэффициент
регуляризации
 α - шаг градиентного спуска

Alternating Least Squares (ALS)

Веса матрицы X фиксируются и происходит обновление весов матрицы Y , затем следует аналогичный шаг обновления весов матрицы X .



$$R \approx XY^T$$
$$\sum_{(u,i) \in R} (r_{ui} - \langle x_u, y_i \rangle)^2 + \lambda (||x_u||^2 + ||y_i||^2) \rightarrow \min$$

Alternating Least Squares (ALS)

Обновление весов происходит путем поиска точного решения.

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

Шаги повторяются до достижения сходимости.

Alternating Least Squares for Implicit Feedback (iALS)

$$\sum_{(u,i)} c_{ui} (p_{ui} - \langle x_u, y_i \rangle)^2 + \lambda (||x_u||^2 + ||y_i||^2) \rightarrow \min$$

где p_{ui} - индикатор наличия фидбека

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

c_{ui} характеризует степень уверенности в том, что объект понравился пользователю

$$c_{ui} = 1 + \alpha r_{ui}$$

Обновление весов происходит аналогично ALS.

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

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

Pros & Cons

Достоинства:

- Сильные бейзлайны “из коробки”
- Быстрое получение предсказаний при помощи ANN
- Truncated SVD, ALS/iALS: возможно получать предсказания, учитывая новые действия пользователя
- ALS/iALS: можно выполнять параллельные вычисления
- Funk SVD: можно кастомизировать функцию потерь

Недостатки:

- Не используются признаки и контекст
- Смещение в сторону рекомендации популярных айтемов
- ALS/iALS: только MSE функция потерь

Вопросы?



Ставим “+”,
если вопросы есть



Ставим “-”,
если вопросов нет



Рефлексия

Цели вебинара

Проверка достижения целей

1. Рассмотрели метод матричной факторизации
2. Узнали, как использовать SVD разложение в задаче рекомендации
3. Познакомились с другими популярными моделями матричной факторизации

Спасибо за внимание!

Приходите на следующие вебинары



Вероника Иванова

DS RecSys

Контакты: @nica_vera