

# Исследование возможности встраивания контекстной информации в алгоритмы коллаборативной фильтрации на основе матричных разложений

Стеценко М. А.    Игнатов Д. И.

Национальный исследовательский университет «Высшая школа экономики»  
Факультет компьютерных наук  
Отделение Прикладной математики и информатики

Москва, 2015

# Оглавление

- ① Введение
- ② Глава 1  
SVD
- ③ Глава 2  
Контекстная информация в CF
- ④ Заключение
- ⑤ Литература

## Аннотация

Наличие контекстной информации является одним из важнейших факторов для построения личных рекомендаций. Однако, классические алгоритмы коллаборативной фильтрации, основанные на матричных разложениях, таких как SVD разложение, используют только информацию о пользователях и предметах и не предоставляют явных методов включения дополнительных факторов. В данной работе будет показан один из методов встраивания контекстной информации в алгоритм, использующий SVD разложение. Для тестирования рассматриваемого метода будет использоваться открытый банк данных MovieLens. База данных содержит пользователей портала MovieLens, каждый из которых оценил не менее 20 фильмов, а так же информацию о каждом фильме.

Context has always been an important factor in personalized Recommender systems. However, standard collaborative filtering algorithms based on matrix factorization rely mainly on user and subject information and don't provide any methods for encapsulating extra data. This work demonstrates such a method based on SVD decomposition. To test results an open data base taken from MovieLens is used. The database provides information about users and movies.

# Введение

- **Рекомендательные системы** – это модели, которые лучше вас знают, что вам хочется.

# Введение

- **Рекомендательные системы** – это модели, которые лучше вас знают, что вам хочется.
  - Netflix - аренда фильмов.

- **Рекомендательные системы** – это модели, которые лучше вас знают, что вам хочется.
  - Netflix - аренда фильмов.
  - Amazon - лидер в области E-Commerce.

- **Рекомендательные системы** – это модели, которые лучше вас знают, что вам хочется.
  - Netflix - аренда фильмов.
  - Amazon - лидер в области E-Commerce.
  - Яндекс.Музыка - стриминг музыки.



- **Рекомендательные системы** – это модели, которые лучше вас знают, что вам хочется.
  - Netflix - аренда фильмов.
  - Amazon - лидер в области E-Commerce.
  - Яндекс.Музыка - стриминг музыки.
- Для множества пользователей  $User$  и предметов  $Objects$  составляем матрицу, каждый элемент матрицы  $a_{ij}$  хранит оценку пользователя  $i$  для предмета  $j$ . Оценки может и не быть, зачастую такие матрицы очень разрежены, много нулей. Теперь задача заключается в заполнении пропусков.

- **Рекомендательные системы** – это модели, которые лучше вас знают, что вам хочется.
  - Netflix - аренда фильмов.
  - Amazon - лидер в области E-Commerce.
  - Яндекс.Музыка - стриминг музыки.
- Для множества пользователей  $User$  и предметов  $Objects$  составляем матрицу, каждый элемент матрицы  $a_{ij}$  хранит оценку пользователя  $i$  для предмета  $j$ . Оценки может и не быть, зачастую такие матрицы очень разрежены, много нулей. Теперь задача заключается в заполнении пропусков.
- **Коллаборативная фильтрация** - метод построения рекомендаций, основываясь на оценках других пользователей.

- Множество переменных, которые влияют на отношение пользователя к предмету и следовательно на рекомендации для этого пользователя, называется **контекстом**. Например, жанр фильма, время покупки товара, темп музыки.
- **Цель работы:** научиться использовать контекстные данные для улучшения результатов работы рекомендательной системы.

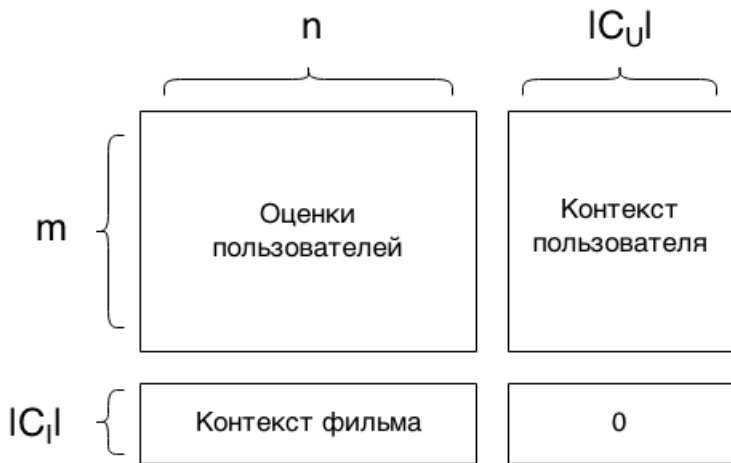
- Типы рекомендательных систем
  - *Content-based Filtering* - используются свойства и содержание рекомендуемых объектов
  - *Collaborative Filtering* - основан на большом количестве собранных от пользователей отзывов
- Способы реализации CF
  - *Метод соседей (Neighborhood based)*
  - *Матричное разложение* - уменьшаем размер матрицы отзывов, путем поиска матриц меньшего размера, которые в произведении дают исходную матрицу

*SVD (Singular Value Decomposition) -  $M_{m,n} = U_{m,f} K_{f,f} I_{n,f}$*

$M$  - это исходная матрица оценок ( $m = |\mathcal{U}|$ ,  $n = |\mathcal{I}|$ ). Элемент матрицы  $U$  содержит веса каждого из факторов для конкретного пользователя. Тоже самое, но для предметов, содержит матрица  $I$ . Матрица  $K$  диагональная, на ее диагонали в убывающем порядке, находятся сингулярные числа матрицы  $M$ .

- *Контекстная информация* - любая информация о предмете или пользователе (пол, возраст, жанр фильма)
- Встраивания контекстной информации
  - **Contextual Pre-Filtering.** Фильтрация данных на входе
  - **Contextual Post-Filtering.** Фильтрация результатов
  - **Contextual Modeling.** Контекстная информация встраивается на уровне модели

## Контекстная информация в CF



## Реализация на Python

- FunkSVD - алгоритм поиска SVD разложения, основанный на градиентном спуске

$$U_{f_i} = U_{f_i} + L(2eI_{f_i} - GU_{f_i})$$

$$I_{f_i} = I_{f_i} + L(2eU_{f_i} - GI_{f_i})$$

- Оценка эффективности

$$\text{Root Mean Square Error (RMSE)} = \sqrt{\frac{1}{n} \sum_i^n (r_i - \hat{r}_i)^2}$$

$$\text{Mean Absolute Error (MAE)} = \sum_i^n |r_i - \hat{r}_i|$$

- Исходный код доступен на [GitHub](#)



## Заключение и результаты

Были получены следующие значения ошибок:

	RMSE	MAE
SVD	0.934347	0.758373
SVD + Context	0.919411	0.753776

В ходе работы был найден способ встраивания контекстной информации в алгоритм коллаборативной фильтрации на основе матричного разложения SVD. Предложенный подход реализован на языке Python. В ходе тестирования было выявлено, что наличие контекстной информации улучшает ключевые метрики, отвечающие за точность рекомендаций.

## Литература I



Faris Alqadah и др. "Biclustering neighborhood-based collaborative filtering method for top-n recommender systems". В: *Knowledge and Information Systems* (авг. 2014). ISSN: 0219-1377. DOI: [10.1007/s10115-014-0771-x](https://doi.org/10.1007/s10115-014-0771-x). URL: <http://link.springer.com/10.1007/s10115-014-0771-x>.



Simon Funk. *Netflix Update: Try This at Home*. URL: <http://sifter.org/~simon/journal/20061211.html>.



Dmitry I Ignatov и др. *Boolean Matrix Factorisation for Collaborative Filtering : An FCA-Based Approach*.



Dmitry I Ignatov и др. *Recommendation of Ideas and Antagonists for Crowdsourcing Platform Witology*.

## Литература II



“Multiverse Recommendation: N-dimensional Tensor Factorization for Context-aware Collaborative Filtering”. В: *ACM Recommender Systems* (2010). URL: <http://xavier.amatriain.net/pubs/karatzoglu-recsys-2010.pdf>.



Mark Pilgrim. “Dive into python 3”. В: *Dive Into Python 3* (2009), с. 1—360. ISSN: 1464-3553. DOI: [10.1007/978-1-4302-2416-7](https://doi.org/10.1007/978-1-4302-2416-7).



Nava Tintarev и Judith Masthoff. *Recommender Systems Handbook*. Т. 54. 2011, с. 479—510. ISBN: [978-0-387-85820-3](https://doi.org/10.1007/978-0-387-85820-3). DOI: [10.1007/978-0-387-85820-3](https://doi.org/10.1007/978-0-387-85820-3). URL: <http://www.springerlink.com/index/10.1007/978-0-387-85820-3>.

Спасибо за внимание!

[makazone@gmail.com](mailto:makazone@gmail.com)