

# Автоматическое дополнение музыкальных плейлистов в рекомендательной системе

Кислинский Вадим Геннадьевич

Московский физико-технический институт

*Курс: Численные методы обучения по прецедентам  
(практика, В. В. Стрижов)/Группа 574, весна 2018*

## Цель

Исследовать метод матричной факторизации в задаче автоматического дополнения плейлистов

## Проблемы

- Классический метод матричной факторизации не учитывает дополнительную информацию о плейлистах и треках
- Проблема холодного старта для новых плейлистов

## Было предложено

Использовать метод совместной матричной факторизации

## Обзор основных подходов к задаче автоматического дополнения плейлистов

- 1 Geoffray Bonnin, Dietmar Jannach. Automated Generation of Music Playlists: Survey and Experiments. ACM Computing Surveys (CSUR). 2014

## Основные методы в задаче top-N рекомендаций

- 2 Paolo Cremonesi al. *Performance of recommender algorithms on top-n recommendation tasks*. 2010

## Метод совместной факторизации

- 3 Dimitrios Rafailidis al. *Modeling the Dynamics of User Preferences in Coupled Tensor Factorization*. 2013

## Алгоритм LCE

- 4 Martin Saveski al. *Item Cold-Start Recommendations: Learning Local Collective Embeddings*. 2014

# Постановка задачи

## Дано

$\mathcal{P}$  - множество из  $n$  плейлистов,  $\mathcal{T}$  - множество из  $m$  треков, матрица  $\mathbf{R} \in \mathbb{R}^{n \times m}$ , где  $\mathbf{R}_{ij} = 1$ , если  $t_j \in p_i$ . Каждый трек описывается автором и альбомом, плейлист названием. Эта информация задается матрицей  $\mathbf{X} \in \mathbb{R}^{n \times (k+d+l)}$ , первые  $k$  столбцов показывают, какие авторы входят в плейлист, следующие  $d$ , какие альбомы, последние  $l$  описывают названия плейлистов

## Задача

Для нового плейлиста построить вектор  $\mathbf{r} \in \mathbb{R}^{m \times 1}$ ,  $i$ -ый элемент которого означает насколько трек  $t_i$  подходит данному плейлисту

# Задача совместной матричной факторизации

Ищется разложение матрицы  $\mathbf{R}$  на две матрицы меньшей размерности

$$\mathbf{R} \approx \mathbf{UV}$$

Предполагая, что профили плейлистов зависят от того, какие исполнители, альбомы входят в плейлист, какие названия у плейлистов, запишем

$$\mathbf{X} \approx \mathbf{UH}$$

Приходим к задаче оптимизации:

$$\begin{aligned} \hat{\mathbf{U}}, \hat{\mathbf{V}}, \hat{\mathbf{H}} = \arg \min_{\mathbf{U}, \mathbf{V}, \mathbf{H}} & \alpha \|\mathbf{R} - \mathbf{UV}\|_F^2 + (1 - \alpha) \|\mathbf{X} - \mathbf{UH}\|_F^2 + \\ & \lambda (\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2 + \|\mathbf{H}\|_F^2), \\ \text{s.t. } & \mathbf{U} \geq 0, \mathbf{V} \geq 0, \mathbf{H} \geq 0 \end{aligned} \quad (1)$$

# Введение дополнительной регуляризации на основе близости плейлистов в пространстве признаков

Составим граф близости плейлистов в котором вершинами будут плейлисты. С помощью матрицы смежности —  $\mathbf{A}$  определим близость, соответственных профилей:

$$\begin{aligned} S &= \frac{1}{2} \sum_{i,j=1}^n \|\mathbf{u}_i - \mathbf{u}_j\|^2 \mathbf{A}_{ij} = \sum_{i=1}^n (\mathbf{u}_i^T \mathbf{u}_i) \mathbf{D}_{ii} - \sum_{i,j=1}^n (\mathbf{u}_i^T \mathbf{u}_j) \mathbf{A}_{ij} = \\ &= \text{Tr}(\mathbf{U}^T \mathbf{D} \mathbf{U}) - \text{Tr}(\mathbf{U}^T \mathbf{A} \mathbf{U}) = \text{Tr}(\mathbf{U}^T \mathbf{L} \mathbf{U}), \end{aligned} \quad (2)$$

Перепишем (1) с учетом  $S$

$$\begin{aligned} \hat{\mathbf{U}}, \hat{\mathbf{V}}, \hat{\mathbf{H}} &= \arg \min_{\mathbf{U}, \mathbf{V}, \mathbf{H}} \alpha \|\mathbf{R} - \mathbf{U}\mathbf{V}\|_F^2 + (1 - \alpha) \|\mathbf{X} - \mathbf{U}\mathbf{H}\|_F^2 + \\ &\quad \beta \text{Tr}(\mathbf{U}^T \mathbf{L} \mathbf{U}) + \lambda (\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2 + \|\mathbf{H}\|_F^2), \\ \text{s.t. } &\quad \mathbf{U} \geq 0, \mathbf{V} \geq 0, \mathbf{H} \geq 0. \end{aligned} \quad (3)$$

Матрицы ( $\mathbf{U}$ ,  $\mathbf{V}$ ,  $\mathbf{H}$ ) обновляются по следующим правилам:

- $\mathbf{U} = \mathbf{U} \odot \frac{\alpha \mathbf{R} \mathbf{V}^T + (1 - \alpha) \mathbf{X} \mathbf{H}^T + \beta \mathbf{A} \mathbf{U}}{\alpha \mathbf{U} \mathbf{V} \mathbf{V}^T + (1 - \alpha) \mathbf{U} \mathbf{H} \mathbf{H}^T + \beta \mathbf{D} \mathbf{U} + \lambda \mathbf{U}}$
- $\mathbf{V} = \mathbf{V} \odot \frac{\alpha \mathbf{U}^T \mathbf{R}}{\alpha \mathbf{U}^T \mathbf{U} \mathbf{V} + \lambda \mathbf{V}}$
- $\mathbf{H} = \mathbf{H} \odot \frac{(1 - \alpha) \mathbf{U}^T \mathbf{X}}{(1 - \alpha) \mathbf{U}^T \mathbf{U} \mathbf{H} + \lambda \mathbf{H}}$

Для нового плейлиста составим строку признакового описания  $\mathbf{x} \in \mathbb{R}^{1 \times (k+d+r)}$ . С помощью метода наименьших квадратов из системы

$$\mathbf{x} = \mathbf{u}\mathbf{H}$$

найдем профиль плейлиста  $\mathbf{u}$ .

$$\mathbf{r} = \mathbf{u}\mathbf{V},$$

где  $i$ -ый элемент вектора  $\mathbf{r}$  означает насколько трек  $t_i$  подходит новому плейлисту, после этого из вектора  $\mathbf{r}$  выбирается top- $N$  значений с индексами  $\{i_1, \dots, i_N\}$  и рекомендуются треки  $\{t_{i_1}, \dots, t_{i_N}\}$ .



Пусть  $R$  список рекомендованных треков,  $G$  - список истинных треков.

$$\text{R-precision} = \frac{|G \cap R_{1:|G|}|}{|G|}$$

$$\text{DCG} = 1 + \sum_{i=2}^{|R|} \frac{\text{rel}_i}{\log_2 i}$$

$$\text{IDCG} = 1 + \sum_{i=2}^{|R|} \frac{1}{\log_2 i}$$

$$n\text{DCG} = \frac{\text{DCG}}{\text{IDCG}}$$

## Данные

Случайная подвыборка из миллиона плейлистов, содержащая 7657 плейлистов и 8560 треков, при этом каждый плейлист содержит не менее пяти треков.

## Базовый алгоритм - PureSVD

$$R = U\Sigma Q^T$$

$$R = PQ^T, P = U\Sigma = RQ$$

$$\hat{r}_u = r_u QQ^T$$

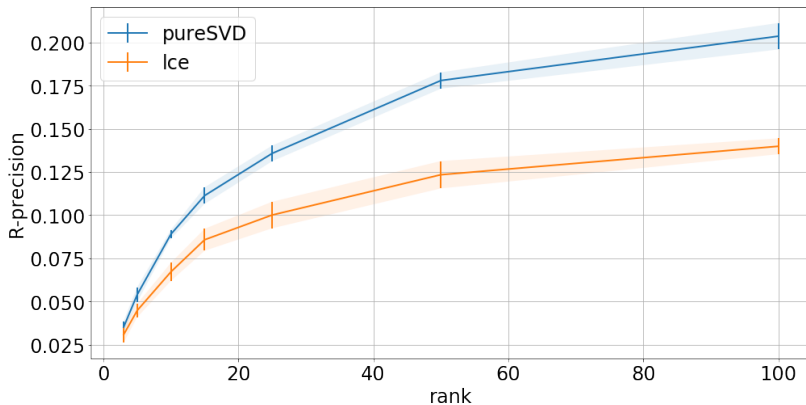


Рис.: Зависимость качества от ранга разложения.

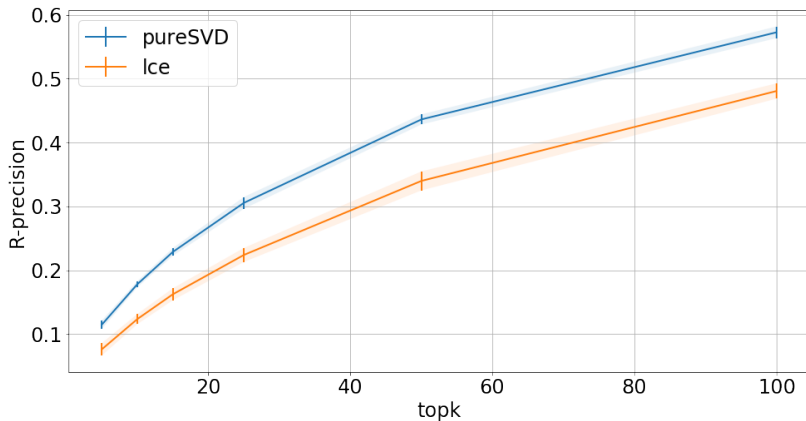


Рис.: Зависимость качества от topk.

# Результаты эксперимента

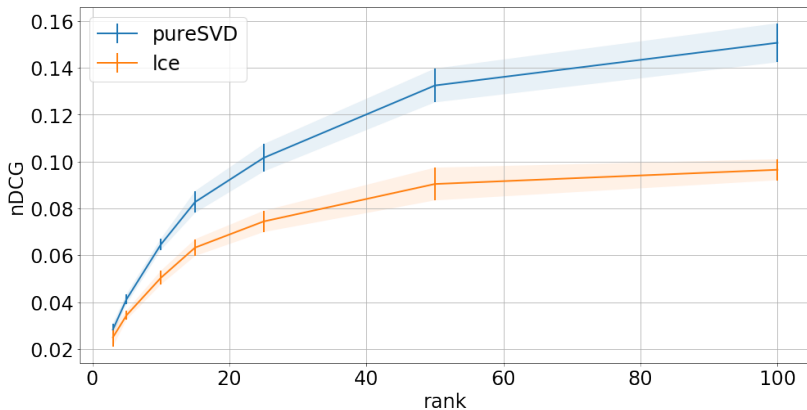


Рис.: Зависимость качества от ранга разложения.

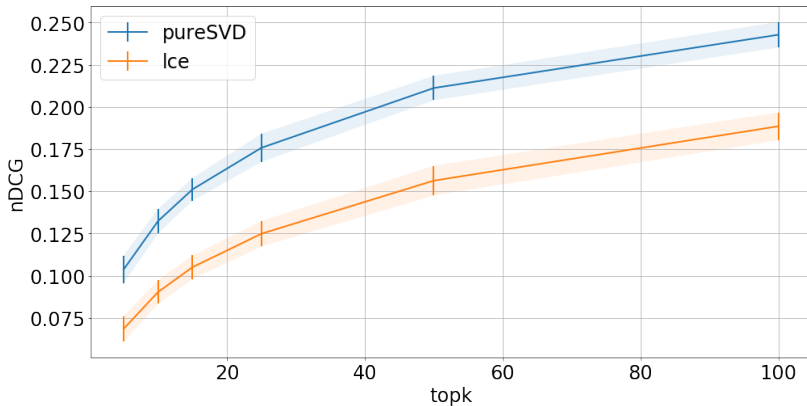


Рис.: Зависимость качества от topk.

- Исследован метод матричной факторизации в задаче автоматического дополнения плейлистов.
- Исследовано влияние учета дополнительной информации в методе матричной факторизации.
- Проверено предположение о близости плейлистов в латентном пространстве.