Автоматическое дополнение плейлистов в рекомендательной системе*

Кислинский В. Г., Фролов Е. , Воронцов К. В.
kislinskiy.vg@phystech.edu; evgeny.frolov@skolkovotech.ru; vokov@forecsys.ru
Московский физико-технический институт

Работа посвящена исследованию метода совместной матричной факторизации в задаче top-N рекомендаций для автоматического продолжения плейлистов. Предлагается модель матричной факторизации, учитывающая дополнительную информацию о плейлистах и треках. Данный метод будет иметь, не только преимущество алгоритмов коллаборативной фильтрации, которые способны выявить скрытые свойства пользователей и обьектов, но также сможеть учитывать контекстную информацию. В данном методе будет введена дополнительная регуляризация, основанная на предположение, что если обьекты близки в пространстве признаков, то они также близки и в латентном факторном пространстве. Для анализа качества представленного алгоритма проводятся эксперименты на выборке из миллиона плейлистов MPD.

Ключевые слова: *задача top-N рекомендаций, совместная матричная факторизация, алгоритм LCE, латентное факторное пространство, коллаборативная фильтрация.*

1 Введение

Существует два основных подхода к задаче автоматического продолжения плейлистов: музыкальный информационный поиск и рекомендательные системы[1]. Рекомендательные системы имеют несколько основных техник, которые можно разделить на две основные группы: коллаборативная фильтрация и content-based подход. Content-based системы используют свойства объектов, создают профили пользователей, опираясь на свойства объектов, которые пользователь предпочел ранее, и рекомендуют новым пользователям наиболее подходящии к их профилю объекты. Коллаборативная фильтрация использует историю действия пользователей в системе для получения новых рекомендаций. Методы коллаборативной фильтрации оценивают сходство пользователей, сходство объектов на основе действий пользователей в прошлом, а потом строят рекомендации. Такие методы обладают хорошей устойчивостью и способны выявлять скрытые свойства объектов, что улучшает релевантность рекомендаций. Основным недостатком этого подхода является проблема холодного старта - неспособность строить рекомендации для новых пользователей и рекомендовать новые объекты.

В данной работе предполагается, что использование свойств объектов и пользователей улучшит качество рекомендаций, получаемых методами коллаборативной фильтрации и решит проблему холодного старта. Будет исследована модель совместной матричной факторизации, подобный подход предлагается в работе[2], где матрица объект-признак факторизуются вместе с матрицей объект-пользователь. Существуют различные техники учета дополнительной информации в модели матричной факторизации, введение дополнительной регуляризации[3], совместная факторизация с матрицами похожести объектов и пользователей[4].

Работа выполнена при финансовой поддержке РФФИ, проект № 00-00-00000. Научный руководитель: Воронцов К. В. Консультант: Фролов Е.

Основной целью работы является решение проблемы холодного старта для пользователей. В задаче автоматического дополнения плейлистов пользователи это плейлисты, обьекты - треки. Для плейлистов, которые содержат достаточно больший набор треков, хорошо работают методы коллаборативной фильтрации, но для новых плейлистов, которые пока содержат мало треков или не содержат совсем, коллаборативная фильтрация не может построить релевантные рекомендации. Если же строить рекомендации на основе свойств плейлистов и треков, то для новых плейлистов можно получить более подходящие рекомендации, например, был создан плейлист с именем "рок" , тогда будет логичнее предложить наиболее популярные треки из рок-музыки, а не что-то другое, но такой content-based подход будет строить слишком тривиальные рекомендации для плейлистов, которые уже имеют много треков. Совместная матричная факторизация объединяет в себе оба этих подхода, будут находиться такие профили плейлистов, которые зависят не только от матрицы плейлист-трек, но и от матрицы плейлист-признак. Также будет рассмотрено предположение о том, что из близости плейлистов в пространстве признаков, следует близость в латентном факторном пространстве, пространстве профилей, для проверки этого предположения будет введена дополнительная регуляризация[2].

2 Постановка задачи

Задано множество треков $\mathcal{T} = \{t_i\}_{i=1}^m$ и множество плейлистов $\mathcal{P} = \{p_i\}_{i=1}^n$,
где $p_i \in \mathcal{T}, |p_i| << |T|$. Каждый трек описывается исполнителем и альбомом, обозначим $\mathcal{S} = \{s_i\}_{i=1}^k$ - множество исполнителей, $\mathcal{L} = \{l_i\}_{i=1}^d$ - множество альбомов, а каждый плейлист имеет название, $\mathcal{A} = \{a_i\}_{i=1}^r$ - множество названий. Определим
матрицу $\mathbf{R} \in \mathbb{R}^{n \times m}$, следующим образом $\mathbf{R}_{ij} = 1$, если $t_j \in p_i$, иначе ноль, также зададим бинарную матрицу $\mathbf{X} \in \mathbb{R}^{n \times (k+d+r)}$, которая описывает треки, каких авторов содержаться в плейлисте - k первых столбцов, каких альбомов d следующих столбцов, имя плейлиста - r последних столбцов. Надо для нового плейлиста p найти N наиболее подходящих треков, для этого будет строиться вектор $\mathbf{r} \in \mathbb{R}^{m \times 1}$, i - ый элемент которого означает насколько трек t_i подходит плейлисту p.

3 Описание метода

Задача матричной факторизации заключается в нахождение двух матриц меньшей размерности, произведение которых приближает исходную. В задаче рекомендаций ищется приближение матрицы рейтингов , в наших терминах $\mathbf{R} \approx \mathbf{UV}$, где $\mathbf{U}$ - матрица профилей плейлистов, $\mathbf{V}$ - матрица профилей треков. Предполагая, что профили плейлистов зависят от того какие исполнители, альбомы входят в плейлист, какие названия у плейлистов, можно записать $\mathbf{X} \approx \mathbf{UH}$, где $\mathbf{H}$ - матрица профилей авторов, альбомов, названий. Таким образом приходим к следующей задачи оптимазации:

$$\mathbf{U}, \mathbf{V}, \mathbf{H} = \underset{\mathbf{U},\mathbf{V},\mathbf{H}}{\arg\min}\, \alpha||\mathbf{R} - \mathbf{UV}||_F^2 + (1-\alpha)||\mathbf{X} - \mathbf{UH}||_F^2 + \lambda(||\mathbf{U}||_F^2 + ||\mathbf{V}||_F^2 + ||\mathbf{H}||_F^2),$$

$$s.t. \mathbf{U} \geqslant 0, \mathbf{V} \geqslant 0, \mathbf{H} \geqslant 0$$

где λ положительный коэффицент регуляризации, а $\alpha \in [0, 1]$, если $\alpha > 0.5$, то матрица U больше зависит от распределения треков по плейлистам, если $\alpha < 0.5$, то наоборот, предполагается, что U зависит от распределения авторов, альбомов по плейлистам и названий плейлистов.

3.1 Введение дополнительной регуляризации на основе близости плейлистов в пространстве признаков

Близость плейлистов в пространстве признаков можно оценивать евклидовым расстоянием. Составим граф близости плейлистов, в котором вершинами будут плейлисты, каждый плейлист соединим ребром с l ближайшими плейлистами, где l ранг разложения матриц $\mathbf{R}$ и $\mathbf{X}$. Матрица $\mathbf{A}$ - матрица смежности графа близости плейлистов. Предполагая, что если плейлисты близки в реальном пространстве признаков, то они также близки в пространстве профилей, с помощью матрицы $\mathbf{A}$ можно определять близость, соответственных профилей:

$$S = \frac{1}{2}\sum_{i,j=1}^{n} ||u_i - u_j||^2 A_{ij} = \sum_{i=1}^{n}(u_i^T u_i)D_{ii} - \sum_{i,j=1}^{n}(u_i^T u_j)A_{ij} = Tr(\mathbf{U}^T\mathbf{D}\mathbf{U}) - Tr(\mathbf{U}^T\mathbf{A}\mathbf{U}) = Tr(\mathbf{U}^T\mathbf{L}\mathbf{U})$$

где $\mathbf{D}$ диоганальная матрица, где на ii месте стоит сумма i строки матрицы $\mathbf{A}$, $\mathbf{L} = \mathbf{D} - \mathbf{A}$. Перепишем (1), учитывая S:

$$\mathbf{U}, \mathbf{V}, \mathbf{H} = \underset{\mathbf{U},\mathbf{V},\mathbf{H}}{\arg\min}\, \alpha||\mathbf{R} - \mathbf{U}\mathbf{V}||_F^2 + (1-\alpha)||\mathbf{X} - \mathbf{U}\mathbf{H}||_F^2 + \beta Tr(\mathbf{U}^T\mathbf{L}\mathbf{U}) + \lambda(||\mathbf{U}||_F^2 + ||\mathbf{V}||_F^2 + ||\mathbf{H}||_F^2)$$

$$s.t. \mathbf{U} \geqslant 0, \mathbf{V} \geqslant 0, \mathbf{H} \geqslant 0$$

3.2 Оптимизационная проблема

Полученная задача оптимизации не является выпуклой по параметрам $(\mathbf{U}, \mathbf{V}, \mathbf{H})$, следовательно поиск глобального минимума затруднительная задача. Предлагается итеративный алгоритм поиска стационарной точки оптимизируемого функционала, который был получен и доказан в статье[2]. Матрицы $(\mathbf{U}, \mathbf{V}, \mathbf{H})$ будут обновляться по следующим правилам:

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

$$\mathbf{V} = \mathbf{V} \odot \frac{\alpha\mathbf{U}^T\mathbf{R}}{\alpha\mathbf{U}^T\mathbf{U}\mathbf{V} + \lambda\mathbf{V}}$$

$$\mathbf{H} = \mathbf{H} \odot \frac{(1-\alpha)\mathbf{U}^T\mathbf{X}}{(1-\alpha)\mathbf{U}^T\mathbf{U}\mathbf{H} + \lambda\mathbf{H}},$$

где $\odot$ - поэлементное умножение, $\dot{-}$ - поэлементное деление.

3.3 Получение рекомендаций

Для того чтобы получить рекомендации для нового плейлиста p составим строку признакого описания $\mathbf{x} \in \mathbb{R}^{1\times(k+d+r)}$, из системы $\mathbf{x} = \mathbf{u}\mathbf{H}$, найдем профиль плейлиста $\mathbf{u}$, метод наименьших квадратов. Теперь умнажая вектор $\mathbf{u}$ на матрицу $\mathbf{V}$, получим вектор $\mathbf{r}$, i-ый элемент, которого означает насколько трек t_i подходит плейлисту p, после этого из вектора $\mathbf{r}$ выбирается top-N значений с индексами $\{i_1, \ldots, i_N\}$ и плейлисту p рекомендуюся треки $\{t_{i_1}, \ldots, t_{i_N}\}$.

4 Оценка качества алгоритма

Качество полученных рекомендаций будет оценивать с помощью двух метрик:
R-precision и $nDCG$ - normalized discounted cumulative gain. Первая из них будет показывать насколько, рекомендованные треки соответсвуют интересам пользователя, вторая

помимо релевантности треков, показывает качество ранжирования треков. Пусть R множество рекомендованных треков, G - множество истинных треков.

R-precision определяется, как отношение правильно рекомендованных треков на количество истинных треков.

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

Чтобы определить $nDCG$, определим сначала DCG и $IDCG$ - идеальный DCG, DCG увеличивается, если релевантно рекомендованные треки находяться выше в списке $top-N$ рекомендаций.

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

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

Теперь определим $nDCG$:

$$nDCG = \frac{DCG}{IDCG}$$

5 Базовый вычислительный эксперимент

Оценка качества алгоритма и сравненение с другими методами будет проводиться на случайной подвыборки из миллиона плейлистов, содержащей 7657 плейлистов и 8560 треков, при этом каждый плейлист содержит не менее пяти треков. По данной подвыборке проводится следующая кроссвалидация: множество плейлистов разбивается на пять примерно равных частей, качество оценивается на каждой из пяти частей поочередно, на остальных обучается, при этом на тестовых плейлистах скрывается часть треков.

Литература

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

[2] Martin Saveski, Amin Mantrach. Item Cold-Start Recommendations: Learning Local Collective Embeddings. Proceeding RecSys '14 Proceedings of the 8th ACM Conference on Recommender systems Pages 89-96. 2014

[3] Yifan Chen, Xiang Zhao. Leveraging High-Dimensional Side Information for Top-N Recommendation. CoRR. 2017

[4] Cold-Start Item and User Recommendation with Decoupled Completion and Transduction Iman Barjasteh, Rana Forsati, Farzan Masrour, Abdol-Hossein Esfahanian, Hayder Radha. Cold-Start Item and User Recommendation with Decoupled Completion and Transduction. 2015