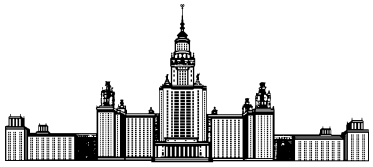

Московский государственный университет имени М. В. Ломоносова
Факультет вычислительной математики и кибернетики
Кафедра математических методов прогнозирования

Борисов Алексей Антонович

Кластеризация сессий в общих учетных записях

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Научный руководитель:
д. ф.-м. н., профессор
Дьяконов Александр Геннадьевич

Москва, 2023

Содержание

Аннотация

Рекомендательные системы являются неотъемлемой частью современного мира, они позволяют среди огромного количества сущностей выбрать наиболее релевантные для конкретного человека. Однако нередко рекомендательной системой с одного аккаунта пользуются одновременно несколько человек. Это может сильно ухудшать качество рекомендаций, так как большинство подходов предполагают, что каждому аккаунту соответствует один реальный пользователь. Целью данного исследования является описание подхода для кластеризации сессий пользовательской активности в совместных аккаунтах, а также проведение необходимых экспериментов для оценки качества представленного подхода. Одной из проблем исследований по данной теме является отсутствие необходимой разметки, однако особенностью данной работы является использование разметки принадлежности действий реальным пользователям на основе распознавания голоса в голосовом помощнике «Маруся». Также в настоящей работе активность пользователя в рекомендательной системе описывается не на уровне отдельных объектов, а на уровне действий, которые пользователь совершает в рекомендательной системе. Такой подход должен лучше отражать желания пользователя.

1 Введение

В современном мире нас окружает огромное количество стриминговых сервисов, например, «Youtube» и «Кинопоиск» — онлайн-сервисы видео, «Яндекс Музыка» и «VK Музыка» — онлайн-сервисы музыки. Во многих таких сервисах распространено явление, когда одним аккаунтом в действительности пользуются несколько человек. Это связано прежде всего с тем, что внутри одной семьи все пользуются сервисом из-под одного аккаунта. Особенно характерно такое использование голосовых помощников: в доме находится одна умная колонка с голосовым помощником, через которую может осуществляться использование стриминговых сервисов, например, онлайн-кинотеатра или сервиса для прослушивания музыки.

У подавляющего большинства стриминговых сервисов имеется своя рекомендательная система, которая предоставляет для каждого пользователя персонализированные релевантные предложения, будь то фильмы или музыка. Однако рекомендательные системы зачастую предполагают, что каждому аккаунту соответствует один реальный пользователь, и возможна ситуация, что рекомендации для совместных аккаунтов будут релевантны только для одного из пользователей или вовсе не будут релевантны никому.

Для борьбы с проблемами, связанными с совместными аккаунтами, необходимо уметь разделять активность внутри аккаунта по реальным пользователям. В работе Jiang et al. [7] выделяют три основные задачи:

1. по истории сессий аккаунта определить набор реальных пользователей;
2. по заданной новой сессии определить, какому пользователю из заранее определенного множества она принадлежит;
3. использование информации о принадлежности сессий разным пользователям для улучшения качества рекомендаций.

Данная работа сосредоточена на исследовании задачи 1, кластеризации сессий пользовательской активности. Ниже в текущем разделе описаны существующие подходы к обработке совместных аккаунтов в рекомендательных системах, а также описывается формальная постановка задачи. В разделе 2 описана общая модель для решения поставленной задачи, а также рассмотрены различные варианты реали-

зации представленной модели. Далее в разделе 3 проанализировано качество рассмотренных решений на реальных данных стримингового сервиса музыки в голосовом помощнике «Маруся». Работы по данной теме сталкиваются с принципиальной проблемой отсутствия необходимой разметки принадлежности действий конкретным пользователям, что мешает реально оценить качество описанных решений, однако в настоящей работе использована разметка, основанная на распознавании голоса.

1.1 Обзор литературы

Исследований по работе с совместными аккаунтами в рекомендательных системах не так много, так как тема является довольно специфичной.

В наиболее ранней из рассмотренных работ, Zhang et al. [20] предлагают представлять совместные аккаунты в виде объединения линейных подпространств, где каждому отдельному пользователю соответствует одна гиперплоскость. Выписывается невыпуклая оптимизационная задача, которую удается решить при помощи EM-алгоритма или обобщенного метода главных компонент (GPCA). Кроме собственно модели, авторы разработали статистический тест, который позволяет оценить вероятность того, что полученный результат является случайным. В этой работе использовался датасет CAMRa2011 [1], обладающий разметкой для 290 совместных аккаунтов, а также были сформированы синтетические совместные аккаунты на основе Netflix Dataset.

Существует ряд статей, которые улучшают качество рекомендаций на случай совместных аккаунтов без явного поиска разбиения аккаунта по пользователям. Например, в работе Bajaj et al. [3] для улучшения качества рекомендаций в сервисе онлайн телевидения предлагается следующий подход: на основе всей имеющейся статистики просмотров формируются кластеры каналов, после чего с помощью алгоритма Apriori определяются частые комбинации кластеров, которые представляют собой некототрые профили пользователей (Personas). Далее для каждого аккаунта выделяется набор соответствующих ему профилей (возможно пересекающихся), которые впоследствии используются при формировании рекомендаций. Схожий подход использовался в работе Verstrepen и Goethals [14] для улучшения качества коллаборативной фильтрации на случай совместных аккаунтов. Вместо того чтобы определять

оценку нового объекта i как сумму его похожестей на объекты S, уже оцененные пользователем:

$$s_{IB}(S, i) = \sum_{j \in S} sim(j, i) \cdot |KNN(j) \cap \{i\}|$$

предлагается оценивать новый объект по подмножеству объектов, дающему наибольшую оценку, и при этом штрафовать за размер использованного подмножества:

$$s_{DAMIB}(S, i) = \max_{D \in 2^S} \frac{1}{|D|^p} \cdot s_{IB}(D, i), \quad p \in [0, 1]$$

Этот подход позволяет избежать ситуации, в которой рекомендации не релевантны никому из пользователей аккаунта из-за того, что они имеют разные вкусы.

Отдельно можно выделить исследования, использующие графовые нейронные сети для выявления и обработки совместных аккаунтов. В работе Wen et al. [18] для каждого аккаунта строится ориентированный граф сессий, вершины в котором обозначают объекты v_i, с которыми взаимодействовал пользователь, а наличие ребра $v_i \to v_j$ означает, что в одной из сессий объект v_j шел после v_i. Векторное представление сессии строится путем усреднения векторных представлений объектов сессии, которые формируются с помощью графовой нейросети. Далее для определения векторных представлений реальных пользователей используется механизм внимания (attention). В другой работе Jiang et al. [7] строят уже один неориентированный граф, содержащий вершины объектов, а также мета-информации, например, вершины исполнителей или альбомов. Векторные представления объектов формируются при помощи алгоритма на основе случайного блуждания. Для получения векторного представления сессии используется эвристическое взвешенное среднее, уменьшающее влияние объектов, с которыми во время сессии взаимодействовали несколько раз. Далее для кластеризации сессий используется алгоритм Affinity Propagation [5].

В большинстве исследований, связанных с обработкой совместных аккаунтов, качество решения оценивается на задаче рекомендаций при помощи классических метрик ранжирования: Recall@n, Precision@n, MRR, NDCG и др. [3, 14, 9, 18, 16, 21] В тех работах, где пытаются оценить качество кластеризации действий из-под пользовательского аккаунта, зачастую используют синтетические данные, так как в открытом доступе практически отсутствуют датасеты с наличием необходимой разметки в совместных аккаунтах [20, 7]. Единственный найденный датасет с такой разметкой,

CAMRa2011 [1], обладает ей только для 290 аккаунтов, чего недостаточно для оценки качества решений. Поэтому для оценки качества кластеризации используются синтетические данные, в которых аккаунты представляют собой объединение 2 и более реальных аккаунтов.

1.2 Постановка задачи

Для начала введем ряд обозначений:

U — множество пользовательских аккаунтов.

P_u — множество реальных людей, пользующихся рекомендательной системой из-под аккаунта $u \in U$.

A — множество действий, которые может совершать пользователь внутри рекомендательной системы.

$s = [a_1, \ldots, a_m]$ — сессия, последовательность действий внутри рекомендательной системы, выполненная одним человеком, $a_i \in A$. Зачастую разумно считать, что если действия были выполнены через незначительный промежуток времени, то они были совершены одним человеком.

S_u — множество сессий, совершенных из-под аккаунта $u \in U$.

Задача заключается в кластеризации сессий, совершенных из-под одного аккаунта. Решением задачи является алгоритм f, который для множества сессий $S_u = \{s_1^u, \ldots, s_N^u\}$ выдает принадлежность каждой сессии одному из K кластеров: $\{(s_1^u, k_1), \ldots, (s_N^u, k_N)\}$, где $k_i \in \{0, \ldots, K\}$. Идеальным результатом является получение взаимно однозначного отображения между кластерами и реальными пользователями P_u. В данной работе количество кластеров предполагается известным заранее и равным количеству реальных пользователей.

Таким образом, естественными функционалами качества в данной задаче являются внешние меры оценки качества кластеризации, использующие информацию о распределении объектов по кластерам. Описание конкретных метрик, которые использовались в экспериментах приведено в разделе 3.1.

2 Предлагаемый подход

В данном разделе вначале будет описана общая модель, которая представляет собой последовательность действий, которые решают поставленную задачу. Далее будет описана предметная область музыкального сервиса и представлен простейший вариант модели, после чего описаны улучшения, которые позволяют повысить качество.

2.1 Общая модель

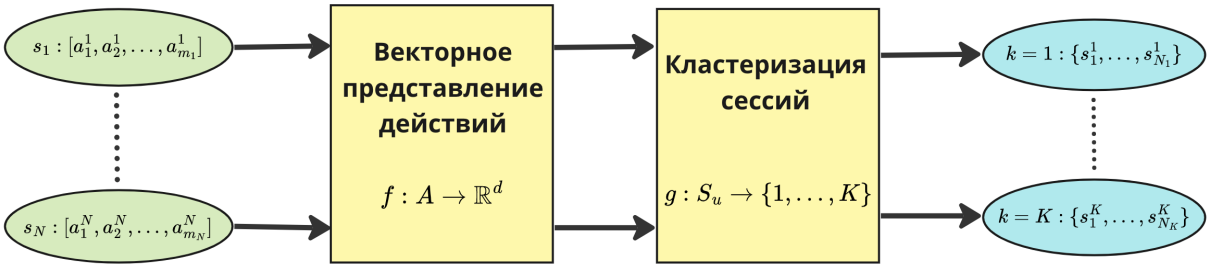

Рис. 1: Общая модель кластеризации сессий одного аккаунта.

На рисунке 1 изображена общая модель кластеризации сессий одного аккаунта, подробнее остановимся на каждом элементе данной схемы.

На вход поступает множество сессий, каждая из которых представляет собой последовательность некоторых действий в рекомендательной системе. Далее для каждого действия строится векторное представление. Несмотря на то, что действия могут быть разного вида, все они должны отображаться в одно векторное пространство, так как следующий шаг работы модели заключается в кластеризации сессий на основе близости векторных представлений действий. Здесь можно выделить три разных подхода:

1. построение векторного представления сессии на основе векторных представлений действий и дальнейшая кластеризация в пространстве сессий;
2. кластеризация в векторном пространстве действий с последующим определением кластера для каждой сессии, например, при помощи голосования среди действий из сессии;

3. кластеризация в векторном пространстве действий с использованием дополнительной априорной информации о принадлежности групп объектов к одному кластеру (действия, относящиеся к одной сессии, обязаны лежать в одном кластере).

2.2 Описание предметной области

Для последующего рассмотрения более конкретных вариантов применения общей модели удобно остановиться на определенной предметной области и описывать предложенные подходы в приложении к ней. Они могут быть естественным образом адаптированы к использованию в других предметных областях. В этой работе рассматривается предметная область стримингового сервиса музыки, интегрированного в голосовой помощник. Примерами таких голосовых помощников являются «Алиса» от компании «Яндекс» и «Маруся» от компании «VK».

Основные сущности: трек, исполнитель и плейлист. Трек — аудиозапись, которая может быть воспроизведена для пользователя. У каждого трека есть информация о его исполнителе. Плейлист — заранее подготовленный список треков, например, в большинстве сервисов понравившиеся треки можно добавить в плейлист «Моя музыка».

В рассматриваемой модели музыкального сервиса выделим три типа действий, которые может совершать пользователь:

- Воспроизведение определенного трека.
  Пример: «Включи We Will Rock You».
- Воспроизведение треков определенного исполнителя.
  Пример: «Включи Pink Floyd».
- Воспроизведение треков определенного плейлиста.
  Пример: «Включи мою музыку», «Включи классику русского рока».

Необходимо предоставить отображение $f : A \to \mathbb{R}^d$, которое каждому действию ставит в соответствие векторное представление. Как уже было упомянуто, все векторные представления должны лежать в одном пространстве.

2.3 Базовое решение

Основным понятием является трек, поэтому сперва рассмотрим варианты получения векторных представлений для треков. Далее они могут быть использованы как основа для построения векторных представлений исполнителей и плейлистов.

Векторные представления треков можно получить, например, при помощи низкоранговых матричных разложений, опишу два основных типа:

- user-item разложение «пользователь на трек», где элементы матрицы отражают факт того, что из-под конкретного пользовательского аккаунта был прослушан конкретный трек, возможно с учетом количества прослушиваний.
- item-item разложение «трек на трек», где элементы матрицы отражают количество сопрослушиваний треков в стриминговом сервисе.

В общем виде задачу низкорангового разложения матрицы можно записать следующим образом:

$$A_{n\times m} \approx W_{n\times k} H^T_{m\times k}$$

Если матрица A представляет собой матрицу взаимодействий n пользовательских аккаунтов с m треками, то строки w_u матрицы W представляют собой векторные представления для пользовательских аккаунтов, а строки h_i матрицы H представляют собой векторные представления для треков.

Если матрица A представляет собой матрицу сопрослушиваний треков между собой (матрица A все равно может являться прямоугольной, так как множества треков слева и справа в разложении могут не совпадать), то строки матриц W и H представляют собой вектора треков, однако в разных векторных пространствах.

Существует большое количество различных способов задать функционал, оптимизируя который можно получить качественное низкоранговое матричное разложение, например:

- Классический квадратичный функционал:

$$\mathscr{L}(W, H) = \sum_{(u,i)\in R} (\langle w_u, h_i\rangle - r_{ui})^2 + \lambda\|W\|_F^2 + \mu\|H\|_F^2$$

- Weighted Regularized Matrix Factorization (WR-MF) [6]:

$$\mathscr{L}(W, H) = \sum_{u,i} c_{ui}(\langle w_u, h_i\rangle - p_{ui})^2 + \lambda\|W\|_F^2 + \mu\|H\|_F^2$$

где $p_{ui} = 1$, если $r_{ui} > 0$, и $p_{ui} = 0$, если $r_{ui} = 0$. c_{ui} отражает уверенность пронаблюдать p_{ui}, например, $c_{ui} = 1 + \alpha r_{ui}$.

- Maximum Margin Matrix Factorization (MMMF) [17]:

$$\mathscr{L}(W, H) = \sum_{(u,i,j)\in D} \max(0, 1 - \langle w_u, h_i - h_j \rangle) + \lambda \|W\|_F^2 + \mu \|H\|_F^2$$

- Bayesian Personalizing Ranking (BPR) [12]:

$$\mathscr{L}(W, H) = \sum_{(u,i,j)\in D} \ln \sigma(\langle w_u, h_i - h_j \rangle) + \lambda \|W\|_F^2 + \mu \|H\|_F^2$$

где R — множество известных оценок «user-item», а D — множество троек (u, i, j), для которых известно, что объект i более релевантен для пользователя u чем объект j.

Представленные функционалы можно оптимизировать, например, при помощи стохастического градиентного спуска (SGD), а также в некоторых частных случаях возможно использование метода чередующихся наименьших квадратов (ALS).

Использование разложения любого из этих типов позволяет получить векторные представления треков, которые можно использовать для действия «воспроизведение трека».

Рассмотрим простейший способ получения векторного представления для исполнителя, которое можно использовать для действия «воспроизведение треков исполнителя». Этот способ заключается в усреднении векторных представлений треков исполнителя:

$$e_{\text{artist}} = \frac{1}{|T_{\text{artist}}|} \sum_{t \in T_{\text{artist}}} e_t, \quad \text{где } T_{\text{artist}} \text{ — множество треков исполнителя artist.}$$

Аналогичным образом можно получить векторное представление для плейлиста, которое можно использовать для действия «воспроизведение треков плейлиста»:

$$e_{\text{playlist}} = \frac{1}{|T_{\text{playlist}}|} \sum_{t \in T_{\text{playlist}}} e_t, \quad \text{где } T_{\text{playlist}} \text{ — множество треков в плейлисте playlist.}$$

Для использования подхода с кластеризацией в векторном пространстве сессии, также необходим способ формирования вектора сессии, по векторам действий. Простейшим способом также является усреднение векторов действий:

$$e_s = \frac{1}{m} \sum_{i=1}^{m} e_{a_i}, \quad \text{где } s = [a_1, \ldots, a_m] \text{ — сессия пользователя.}$$

Таким образом, был описан способ формирования векторов для всех трех типов действий, а также сессий. Далее получить разбиение сессий по пользователям можно, применив к множеству векторных представлений сессий/действий произвольный алгоритм кластеризации, например, K-средних. Отдельный вопрос заключается в выборе меры сходства между векторами. При использовании матричных разложений, наиболее частым выборов являются расстояние Евклида и косинусное расстояние (подчеркнем, что косинусное расстояние не является метрикой, так как для него не выполнено неравенство треугольника).

Также стоит отметить, что для использования алгоритма K-средних с использованием косинусного расстояния достаточно нормировать все вектора и использовать евклидово расстояние, так как таком случае его оптимизация эквивалентна оптимизации косинусного расстояния:

$$\left\| \frac{x}{\|x\|_2} - \frac{y}{\|y\|_2} \right\|_2^2 = \left\langle \frac{x}{\|x\|_2}, \frac{x}{\|x\|_2} \right\rangle - 2 \cdot \left\langle \frac{x}{\|x\|_2}, \frac{y}{\|y\|_2} \right\rangle + \left\langle \frac{y}{\|y\|_2}, \frac{y}{\|y\|_2} \right\rangle = 2 - 2\frac{\langle x, y \rangle}{\|x\|_2 \cdot \|y\|_2}$$

2.4 One-Step-ALS

Рассмотрим подход, позволяющий получить более качественное векторное представление для набора векторов $S = \{e_1, \ldots, e_m\}$ чем простое усреднение. Этот подход можно будет использовать как для получения векторных представлений исполнителей и плейлистов, так и для формирования векторного представления сессии по векторным представлениям действий.

Фактически, представленный ранее способ с усреднением векторов, заключается в минимизации следующего функционала:

$$\mathscr{L}(S) = \frac{1}{m} \sum_{i=1}^{m} \|e_S - e_i\|_2^2 \longrightarrow \min_{e_S}$$

Предлагается воспользоваться другим функционалом, который уже упоминался при описании способов матричного разложения:

$$\mathscr{L}(S) = \frac{1}{m} \sum_{i=1}^{m} (\langle e_S, e_i \rangle - r_i)^2 + \lambda \|e_S\|_2^2 \longrightarrow \min_{e_S}$$

Для представленной задачи минимизации можно записать аналитическое решение:

$$\frac{\partial}{\partial e_S} \mathscr{L}(S) = \frac{2}{m} \sum_{i=1}^{m} (\langle e_S, e_i \rangle - r_i) \cdot e_i + 2\lambda \cdot e_S = 2 \cdot \left(\frac{1}{m} \sum_{i=1}^{m} e_i e_i^T + \lambda \cdot I \right) \cdot e_S - \frac{2}{m} \sum_{i=1}^{m} r_i e_i = 0$$

$$\hat{e}_S = \min_{e_S} \mathscr{L}(S) = \left(\frac{1}{m}\sum_{i=1}^{m} e_i e_i^T + \lambda \cdot I\right)^{-1} \cdot \frac{1}{m}\sum_{i=1}^{m} r_i e_i$$

Использование представленного функционала позволяет получить более качественные векторные представления с точки зрения косинусного расстояния, так как в данном случае оптимизируется скалярное произведение, а не l_2-норма. Если все $r_i = 1$, то с ростом величины коэффициента регуляризации λ направление результирующего векторного представления стремится к направлению вектора полученного простым усреднением векторов e_i:

$$\left(\frac{1}{m}\sum_{i=1}^{m} e_i e_i^T + \lambda \cdot I\right)^{-1} \approx \frac{1}{\lambda} \cdot I, \quad \text{при больших значениях } \lambda.$$

На рисунке 2 представлена визуализация нормированных векторов треков исполнителя *Pink Floyd*, а также векторов самого исполнителя, полученных с помощью усреднения, а также применения шага ALS с различными значениями коэффициента λ. Визуализация была получена путем снижения размерности до 2 с помощью метода главных компонент (PCA). Эта визуализация демонстрирует, что с ростом величины коэффициента регуляризации λ направление результирующего векторного представления стремится к направлению вектора полученного простым усреднением векторов.

Рис. 2: Двумерная проекция на главные компоненты нормированных векторов исполнителей, полученных различными способами.

Можно использовать различные значения r_i, чтобы дать некоторым векторам большее влияние на итоговый вектор набора. Например при формировании векторного представления исполнителя можно давать популярным трекам больший вес, так как они могут лучше характеризовать исполнителя. Если необходимо учитывать все объекты с одинаковым весом, то можно положить все r_t равными единице. Далее

в разделе 2.5 будет представлен альтернативный способ учета важности отдельных элементов набора S.

Естественным названием для данного подхода является One-Step-ALS, так как фактически получение вектора для набора векторов заключается в применении одного шага алгоритма ALS к матрице A «наборы векторов на векторы», где в строке соответствующей набору векторов S стоят значения r_i в столбцах соответствующих векторам $e_i \in S$.

Оценим вычислительную сложность обоих представленных подходов для формирования векторного представления для набора векторов. Используем следующие обозначения: N — количество векторов в наборе, d — размерность векторов. Подход с усреднением векторов имеет сложность $O(N \cdot d)$. При осуществлении одного шага ALS необходимо вычислять N матриц вида $e_i e_i^T$, а также находить обратную матрицу, поэтому итоговая вычислительная сложность данного подхода — $O(N \cdot d^2 + d^3)$.

В представленной работе d это размерность векторов из низкорангового матричного разложения, поэтому d не превосходит нескольких сотен. Количество треков исполнителя, треков в плейлисте или действий в сессии также в подавляющем числе случаев не превосходит нескольких сотен. Таким образом, в данном случае сложность $O(N \cdot d^2 + d^3)$ не представляет больших проблем, тем более что, например, вектора для исполнителей и плейлистов могут быть посчитаны заранее. Однако при использовании одного шага ALS для формирования векторов сессий по векторным представлениям действий это может стать узким местом модели.

2.5 Взвешивание действий

Одним из предложенных способов определения кластера для сессии является кластеризация действий пользователя с последующим голосованием внутри каждой отдельной сессии. При таком подходе, все действия имеют одинаковое влияние на определение итогового кластера, однако в силу того что действия могут иметь разную природу, некоторые из них могут являться намного более важными чем другие. Пример: в онлайн магазине пользователь может совершать следующие действия: «открыть страницу с товаром», «положить товар в корзину», «купить товар». Очевидно,

что покупка является более важным действием чем просмотр товара. Решением данной проблемы является использование взвешенного голосования.

Пусть имеются действия n типов: $A_1, \ldots, A_n$ с весами $w_1, \ldots, w_n$,
где $A_i \subseteq A, w_i \geq 0,\ \forall i = \overline{1,n}, \quad A_i \cap A_j = \emptyset,\ \forall i \neq j$. Тогда для сессии $s = [a_1, \ldots, a_m]$ с известными результатами кластеризации действий $[(a_1, k_1), \ldots, (a_m, k_m)]$ кластер определяется следующим образом:

$$k_s = \arg\max_k \sum_{i=1}^{m} w_{a_i} \cdot [k = k_i], \quad \text{где } w_{a_i} \text{ обозначает вес действия } a_i$$

Аналогичная проблема с неравноценностью действий возникает и в подходе, использующем векторные представления сессий. На данный момент было рассмотрено два способа формирования векторного представления сессии:

1. усреднение векторных представлений действий.
2. использование одного шага алгоритма ALS для набора векторных представлений действий.

Первый подход может быть естественным образом обобщен на случай взвешенного усреднения:

$$e_s = \frac{1}{\sum_{i=1}^{m} w_{a_i}} \sum_{i=1}^{m} w_{a_i} \cdot e_{a_i}$$

Второй подход, One-Step-ALS, как уже было упомянуто в соответствующем разделе также поддерживает возможность учета важности векторов в наборе с помощью варьирования значений r_i, однако также можно рассмотреть следующий взвешенный функционал (для простоты здесь $r_i = 1$):

$$\mathscr{L}(s) = \frac{1}{\sum_{i=1}^{m} w_{a_i}} \sum_{i=1}^{m} w_{a_i} (\langle e_s, e_{a_i} \rangle - 1)^2 + \lambda \|e_s\|^2 \longrightarrow \min_{e_s}$$

Для него имеет место следующее аналитическое решение:

$$\hat{e}_s = \min_{e_s} \mathscr{L}(s) = (\frac{1}{\sum_{i=1}^{m} w_{a_i}} \sum_{i=1}^{m} w_{a_i} e_{a_i} e_{a_i}^T + \lambda \cdot I)^{-1} \cdot \frac{1}{\sum_{i=1}^{m} w_{a_i}} \sum_{i=1}^{m} w_{a_i} e_{a_i}$$

В данной работе веса w_i рассматриваются как гиперпараметры, и могут быть подобраны при помощи стандартных методов оптимизации гиперпараметров: перебора значений по сетке (Grid Search), случайного поиска (Random Search), байесовской оптимизации (Bayesian Optimization) и др.

2.6 Кластеризация с использованием априорной информации

При осуществлении кластеризации в пространстве действий хочется использовать информацию о том, что действия из одной сессии относятся к одному кластеру. Этого можно достичь, например, при помощи агломеративной кластеризации [10].

В агломеративной кластеризации процесс группировки похожих объектов начинается с создания нескольких групп, где каждая из групп содержит один объект, и затем последовательно две наиболее похожие группы объединяются в одну до тех пор пока не останется единая группа. Предлагается использовать следующую модификацию: вместо того чтобы начинать с групп которые содержат по одному объекту, можно начинать с групп, которые представляют собой действия из одной сессии. Таким образом все действия из одной сессии всегда будут принадлежать одному кластеру. Можно считать, что в данном случае производится кластеризация в пространстве сессий, но теперь каждая сессия представляется не одним вектором, а целым множеством векторов действий из этой сессии.

Рассмотрим классические способы подсчета расстояний между группами u и v в агломеративной кластеризации:

- single linkage:
$$d(u,v) = \min_{i,j}(dist(u[i], v[j])),$$
- complete linkage:
$$d(u,v) = \max_{i,j}(dist(u[i], v[j])),$$
- average linkage (UPGMA):
$$d(u,v) = \frac{1}{|u| \cdot |v|} \sum_{i,j} dist(u[i], v[j]),$$
- weighted linkage (WPGMA):
$$d(u,v) = \frac{1}{2}(d(s,v) + d(t,v)),$$
где кластер u был образован кластерами s и t.
- Ward's linkage:
$$d(u,v) = \sqrt{\frac{2|u||v|}{|u|+|v|}} \|c_u - c_v\|_2,$$
где c_s — центроид кластера s.

- centroid linkage (UPGMC):

$$d(u, v) = \|c_u - c_v\|_2,$$

- median linkage (WPGMC):

$$d(u, v) = \|w_u - w_v\|_2,$$

  где w_s это итеративно обновляющаяся «медиана» кластера. Если кластер s образован кластерами s_1 и s_2, то $w_s = \frac{1}{2}(w_{s_1} + w_{s_2})$.

Заранее определить какой из способов окажется наилучшим в настоящей задаче затруднительно, это будет исследовано в разделе 3 в конкретных экспериментах. Можно отметить, что при использовании single linkage, все сессии, содержащие одинаковый трек или исполнителя, будут объединяться в одну группу.

Кроме того, существует проблема, связанная с тем, что алгоритм агломеративной кластеризации не использует информацию о числе кластеров. При выделении фиксированного числа кластеров из иерархии возможна ситуация, в которой некоторые кластеры получаются очень маленького размера (им могут соответствовать, например, некоторые выбросы).

3 Эксперименты

Для исследования возможности решения поставленной задачи, а также оценки качества представленных подходов, были проведены эксперименты на двух датасетах, которые были сформированы на основе логов прослушивания музыки в голосовом помощнике Маруся:

- синтетические данные, в которых совместные аккаунты пользователей сформированы при помощи объединения реальных аккаунтов;
- реальные данные, в которых используются аккаунты, под которыми совершалось много активности как взрослым, так и детским голосом.

Более подробное описание того как формировались данные приведено в разделах 3.2 и 3.3 соответственно. Перед этим в разделе 3.1 кратко описаны внешние метрики оценки качества кластеризации, которые используются в экспериментах.

Вектора треков были сформированы с помощью низкорангового разложения матрицы сопрослушиваний треков в музыкальном сервисе. В качестве алгоритма матричного разложения использовался BPR [12].

В предыдущем разделе было описано большое число вариантов реализации элементов представленной общей модели. Резюмируя, имеем следующую вариативность в конфигурациях моделей, которые будут сравниваться в экспериментах:

- Способ формирования векторов исполнителей и плейлистов:
  - ➤ Усреднение векторов треков.
  - ➤ Использование одного шага ALS на векторах треков.
- Способ получения итоговой кластеризации:
  - ➤ Кластеризация векторных представлений сессий, полученных при помощи: усреднения, одного шага ALS на векторах действий, а также взвешенных версий усреднения и шага ALS (с подбором оптимальных весов).
  - ➤ Кластеризация векторных представлений действий с помощью алгоритма K-средних с последующим голосованием, а также взвешенным голосованием (с подбором оптимальных весов).
  - ➤ Кластеризация векторных представлений действий с помощью агломеративной кластеризации с инициализацией групп объектов действиями из одной сессии (с подбором способа подсчета расстояния между группами).
- Мера близости для кластеризации:
  - ➤ Расстояние Евклида.
  - ➤ Косинусное расстояние.

При использовании одного шага ALS, рассматривались следующие значения коэффициента регуляризации λ: 0.1, 1.0, 10.0, 100.0.

При использовании взвешенного усреднения и шага ALS рассматривались веса, которые суммировались в единицу: $w_{\text{track}} + w_{\text{artist}} + w_{\text{playlist}} = 1$, и подбирались полным перебором по сетке с шагом 0.1.

Подбор гиперпараметров производился на отложенной выборке.

Для оценки качества каждой конфигурации модели при помощи бутстрепа определялись доверительные интервалы с уровнем значимости $\alpha = 0.05$ для средних значений внешних метрик кластеризации.

3.1 Внешние метрики оценки качества кластеризации

В представленных ниже экспериментах используются следующие метрики качества кластеризации [2]:

- Rand Index (RI) — мера, оценивающая какая доля пар элементов, которые находились в одном классе, и тех пар элементов, которые находились в разных классах, сохранили это состояние после кластеризации алгоритмом. Значение 1 соответствует полному совпадению кластеров с истинными классами, 0 соответствует отсутствию совпадений.
- Adjusted Rand Index (ARI) — отнормированная версия Rand Index:

$$\text{ARI} = \frac{\text{RI} - \mathbb{E}[\text{RI}]}{\max(\text{RI}) - \mathbb{E}[\text{RI}]}$$

  Значение 0 соответствует случайному определению кластеров, значение 1 соответствует идеальному совпадению кластеров с истинными классами. Может принимать значения меньше нуля.
- Взаимная информация (Mutual information, MI) — мера, описывающая количество информации, содержащееся в одной случайной величине относительно другой:

$$MI(Y_{\text{true}}, Y_{\text{pred}}) = H(Y_{\text{true}}) - H(Y_{\text{true}}|Y_{\text{pred}}) = H(Y_{\text{true}}) + H(Y_{\text{pred}}) - H(Y_{\text{true}}, Y_{\text{pred}}),$$

  где $H(X)$ — энтропия случайной величины X, $H(X|Y)$ — условная энтропия.
- Скорректированная взаимная информация (Adjusted mutual information, AMI) — вариант нормировки взаимной информации:

$$\text{AMI} = \frac{\text{MI} - \mathbb{E}[\text{MI}]}{\text{mean}(H(Y_{\text{true}}), H(Y_{\text{pred}})) - \mathbb{E}[\text{MI}]}$$

- Полнота (Completeness) — мера, отвечающая за то, что все объекты одного класса принадлежат одному кластеру:

$$\text{Completeness} = 1 - \frac{H(Y_{\text{pred}}|Y_{\text{true})}}{H(Y_{\text{pred}})}$$

- Однородность (Homogeneity) — мера, отвечающая за то, что кластер содержит только объекты одного класса:

$$\text{Homogeneity} = 1 - \frac{H(Y_{\text{true}}|Y_{\text{pred})}}{H(Y_{\text{true}})}$$

- V-Measure — среднее гармоническое полноты и однородности:

$$\text{V-Measure}_\beta = \frac{(1+\beta)\cdot \text{Homogeneity}\cdot \text{Completeness}}{\beta\cdot \text{Homogeneity} + \text{Completeness}}$$

  Вообще говоря, V-Measure с $\beta = 1$ совпадает с нормированной взаимной информацией $NMI = \dfrac{2\cdot MI}{H(Y_{\text{true}}) + H(Y_{\text{pred}})}$.
- BCubed — метрика удовлетворяющая 4 свойствам: Гомогенность, Полнота, «Rag-Bag» и «Cluster size vs Quantity»[2]:

$$\text{Correctness}(x, x') = \begin{cases} 1 & , \ C(x) = C(x') \wedge L(x) = L(x') \\ 0 & , \ \text{otherwise} \end{cases}$$

$$\text{Precision-BCubed} = \text{Avg}_x[\text{Avg}_{x':C(x)=C(x')} Correctness(x, x')]$$

$$\text{Recall-BCubed} = \text{Avg}_x[\text{Avg}_{x':L(x)=L(x')} Correctness(x, x')]$$

$$\text{F-BCubed} = \frac{2\cdot \text{Precision-BCubed}\cdot \text{Recall-BCubed}}{\text{Precision-BCubed} + \text{Recall-BCubed}}$$

3.2 Синтетические данные

Датасет синтетических совместных аккаунтов был сформирован следующим образом: были взяты 10 000 пользовательских аккаунтов, из-под которых было совершено более 600 различных действий за фиксированный отрезок времени. Далее для каждого аккаунта были выделены сессии длины более 10. Два действия принадлежат одной сессии, если между ними прошло не более 15 минут. После этого было случайным образом выбрано 5 000 пар аккаунтов, путем объединения которых были сформированы синтетические совместные аккаунты.

В первую очередь проверим качество кластеризации при рассмотрении каждого действия в отдельности (без учета сессий). В такой постановке задачи, нет необходимости проводить последующее голосование или формирование вектора сессии, так как каждое действие представляет собой отдельную «сессию». В таблице 1 приведены соответствующие результаты для двух способов формирования векторов исполнителей и плейлистов, а также разных способов оценки расстояний. Кроме того, для сравнения указаны значения метрик, соответствующие случайной кластеризации.

	ARI	AMI	V-Measure	BCubed
Случайная кластеризация	0.0	0.0	0.0	0.5206 ± 0.0007
K-средних, усреднение, расстояние Евклида	0.3195 ± 0.0072	0.3107 ± 0.0057	0.3109 ± 0.0059	0.7051 ± 0.0030
K-средних, усреднение, косинусное расстояние	0.3189 ± 0.0070	**0.3150 ± 0.0058**	**0.3152 ± 0.0059**	0.7065 ± 0.0029
K-средних, шаг ALS, косинусное расстояние	0.3173 ± 0.0071	**0.3148 ± 0.0060**	**0.3150 ± 0.0059**	0.7062 ± 0.0030

Таблица 1: Результаты кластеризации без учета сессий.

Как мы видим, в данном случае результаты практически не отличаются для разных конфигураций модели. Однако можно отметить, что использование косинусного расстояния дает незначительный выигрыш в метриках AMI и V-Measure.

Дальше по очереди рассмотрим три способа получения итоговой кластеризации с учетом информации о принадлежности действий одной сессии.

В таблице 2 приведены результаты, полученные при использовании кластеризации действий с последующим голосованием. Также рассматривались два способа формирования векторов исполнителей и плейлистов, два способа подсчета расстояний, а также применение взвешенного голосования.

Использование расстояния Евклида в данном случае дает существенно большие значения метрик, а также использование шага ALS снова не дает значимого преимущества над простым усреднением при использовании косинусного расстояния. При этом для любой конфигурации использование взвешенной версии позволяет увеличить качество. Стоит отметить, что веса действий подбирались на отложенной выборке, и оптимальные веса получились следующие: $w_{\text{track}} = 0.2$, $w_{\text{artist}} = 0.4$, $w_{\text{playlist}} = 0.4$. Тот факт, что веса для исполнителей и плейлистов получились больше можно объяснить следующим образом: запросы на прослушивание определенного

	ARI	AMI	V-Measure	BCubed
усреднение, расстояние Евклида, невзвешенная версия	0.3521 ± 0.0077	0.3568 ± 0.0068	0.3625 ± 0.0067	0.7237 ± 0.0032
усреднение, косинусное расстояние невзвешенная версия	0.3460 ± 0.0077	0.3530 ± 0.0067	0.3588 ± 0.0066	0.7217 ± 0.0033
шаг ALS, косинусное расстояние невзвешенная версия	0.3449 ± 0.0078	0.3533 ± 0.0069	0.3592 ± 0.0066	0.7219 ± 0.0032
усреднение, расстояние Евклида, взвешенная версия	**0.3594 ± 0.0079**	**0.3620 ± 0.0067**	**0.3677 ± 0.0066**	**0.7260 ± 0.0033**
усреднение, косинусное расстояние взвешенная версия	0.3520 ± 0.0078	0.3565 ± 0.0068	0.3623 ± 0.0068	0.7230 ± 0.0033
шаг ALS, косинусное расстояние невзвешенная версия	0.3514 ± 0.0078	0.3574 ± 0.0068	0.3631 ± 0.0067	0.7235 ± 0.0032

Таблица 2: Результаты с использованием подхода с кластеризацией действий и последующим голосованием.

трека более «шумные», чем запросы на прослушивание треков определенного исполнителя или плейлиста, поэтому при определении реального пользователя для сессии нужно больше обращать внимания именно на факт прослушивания треков определенного исполнителя или плейлиста.

Теперь рассмотрим подход с формированием векторных представлений сессий и их последующей кластеризацией. Как можно видеть из предыдущих экспериментов, использование шага ALS при формировании векторных представлений исполнителей и плейлистов не дает значимого результата, однако в данном случае шаг ALS применяется для формирования вектора сессии по векторам действий. Также рас-

сматривается использование взвешенного усреднения и взвешенного ALS. Результаты приведены в таблице 3.

	ARI	AMI	V-Measure	BCubed
усреднение, расстояние Евклида, невзвешенная версия	0.4315 ± 0.0082	0.4341 ± 0.0071	0.4391 ± 0.0070	0.7599 ± 0.0032
усреднение, косинусное расстояние невзвешенная версия	0.4670 ± 0.0083	0.4484 ± 0.0074	0.4532 ± 0.0074	0.7666 ± 0.0036
шаг ALS, расстояние Евклида, невзвешенная версия	**0.4939 ± 0.0086**	**0.4715 ± 0.0075**	**0.4761 ± 0.0076**	**0.7779 ± 0.0076**
шаг ALS, косинусное расстояние невзвешенная версия	0.4870 ± 0.0085	0.4686 ± 0.0077	0.4733 ± 0.0074	0.7760 ± 0.0036
усреднение, расстояние Евклида, взвешенная версия	0.4327 ± 0.0082	0.4339 ± 0.0070	0.4380 ± 0.0071	0.7592 ± 0.0033
усреднение, косинусное расстояние взвешенная версия	0.4656 ± 0.0086	0.4451 ± 0.0075	0.4521 ± 0.0074	0.7650 ± 0.0035
шаг ALS, расстояние Евклида, взвешенная версия	**0.4931 ± 0.0085**	**0.4699 ± 0.0076**	**0.4746 ± 0.0076**	**0.7773 ± 0.0036**
шаг ALS, косинусное расстояние взвешенная версия	0.4846 ± 0.0084	0.4652 ± 0.0076	0.4700 ± 0.0073	0.7746 ± 0.0036

Таблица 3: Результаты с использованием подхода с формированием векторных представлений сессий и их последующей кластеризацией.

В данном случае использование шага ALS для формирования векторов сессий дает очень значительный прирост в качестве. Расстояние Евклида позволяет получить более высокие значения метрик чем косинусное расстояние при использовании шага ALS, однако при усреднении векторов действий косинусное расстояние дает существенно лучшие результаты. При этом использование весов для различных действий при формировании вектора сессии не дает значимого преимущества, в данном случае также использовались веса $w_{\text{track}} = 0.2$, $w_{\text{artist}} = 0.4$, $w_{\text{playlist}} = 0.4$.

Осталось рассмотреть подход с использованием агломеративной кластеризации над векторными представлениями действий с инициализацией групп с помощью действий из одной сессии. Как уже упоминалось в разделе 2.6 при использовании агломеративной кластеризации возникает проблема с последующим выделением фиксированного количества кластеров из результатов иерархической кластеризации, так как один из кластеров может оказаться очень маленького размера вплоть до 1-2 элементов. Использование Ward's linkage в свою очередь позволяет в большинстве случаев избежать этой проблемы и получать качественное разделение на 2 кластера. Именно эта мера расстояния между группами и была использована в экспериментах, результаты которых представлены в таблице 4.

	ARI	AMI	V-Measure	BCubed
усреднение, расстояние Евклида	0.3551 ± 0.0080	0.3806 ± 0.0067	0.3862 ± 0.0067	0.7371 ± 0.0032
усреднение, косинусное расстояние	**0.3743 ± 0.0077**	**0.3907 ± 0.0066**	**0.3962 ± 0.0065**	**0.7431 ± 0.0030**
шаг ALS, косинусное расстояние	0.3644 ± 0.0077	0.3861 ± 0.0065	0.3917 ± 0.0065	0.7421 ± 0.0030

Таблица 4: Результаты с использованием алгомеративной кластеризации с инициализацией групп действиями из сессий.

Таким образом, лучшее достигнутое значение ARI без использования информации о принадлежности действий одной сессии (не учитывается время между действиями) — **0.3195**. Подход с использованием голосования по действиям из одной сессии позволил увеличить это значение до **0.3594**. Используя альтернативный подход с

использованием агломеративной кластеризации, удалось достигнуть ARI равного **0.3743**, а наилучшее качество было получено при помощи подхода с формированием векторного представления сессии с помощью одного шага ALS — **0.4939**.

Как показали эксперименты, представленные модификации модели, а именно: добавление весов для действий и использование шага ALS вместо усреднения, позволяют в некоторых случаях повысить качество.

3.3 Разметка по голосу

При формировании второго датасета, использовались аккаунты, из-под которых совершалось достаточно большое количество действий как при использовании взрослого, так и детского голоса. В датасет попали аккаунты, из-под которых было совершено ≥ 200 действий как детским голосом, так и взрослым, и при этом доля каждого голоса не превосходит 75%.

Для каждого такого аккаунта были выделены сессии длины не менее 10, в которых время между двумя последовательными действиями не превосходит 15 минут. При этом возможна ситуация, что в сессии присутствуют действия, которые были совершены разными пользователями. В таком случае истинная метка сессии соответствовала тому пользователю, доля действий которого была больше. На рисунке 3 изображена гистограмма доли действий детским голосом в сессиях из датасета (0 соответствует тому, что все действия были совершены взрослым голосом, а 1 тому, что все действия были совершены детским голосом). Можно сделать вывод, что для подавляющего числа сессий «владелец» определяется однозначно.

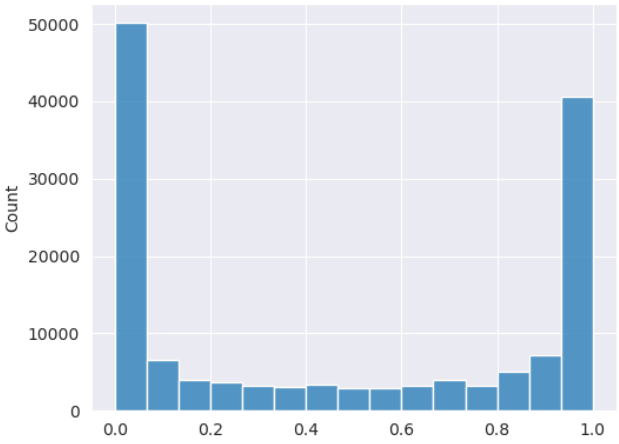

Рис. 3: Распределение доли действий детским голосом среди сессий из рассматриваемого датасета.

В таблице 5 представлены результаты кластеризации сессий совместных аккаунтов с помощью разных подходов в конфигурациях, которые показали наилучшее качество на синтетических данных, а именно:

1. Кластеризация действий с помощью алгоритма K-средних без учета времени между действиями, расстояние Евклида.
2. Кластеризация действий с последующим взвешенным голосованием по действиям из сессии, расстояние Евклида.
3. Кластеризация векторных представлений сессий, полученных при помощи шага ALS, расстояние Евклида.
4. Кластеризация действий при помощи агломеративной кластеризации, Ward's linkage, косинусное расстояние.

Во всех описанных конфигурациях используются вектора исполнителей и плейлистов, полученные с помощью усреднения векторов треков.

Подход	ARI	AMI	V-Measure	BCubed
1	0.1137 0.0055	0.1172 0.0045	0.1181 0.0045	0.6157 0.0025
2	0.1373 0.0070	0.1532 0.0064	0.1721 0.0061	0.6442 0.0031
3	**0.2270 0.0089**	**0.2187 0.0080**	**0.2350 0.0079**	**0.6685 0.0039**
4	0.1570 0.0081	0.1811 0.0071	0.2061 0.0067	0.6861 0.0027

Таблица 5: Результаты применения различных подходов на датасете с разметкой по голосу.

Ожидаемо результаты получились хуже чем при использовании синтетических данных, так как в реальности редко бывает такое, что члены одной семьи слушают абсолютно разные треки. Зачастую у них пересекаются вкусы, из-за чего произвести достаточно качественную кластеризацию становится невозможно.

Как и на синтетическом датасете лучший результат показал подход с построением векторного представления сессии, а худший результат показал подход с кластеризацией действий с последующим голосованием (за исключением подхода, в котором вообще никак не использовалась информация о сессиях).

3.4 Проблема нескольких паттернов потребления

При анализе экспериментов, было замечено, что распределение метрик кластеризации по отдельным аккаунтам имеет странный вид. Значения из отрезка [0.1, 1.0] достигаются примерно в одинаковом числе случаев, а также наблюдается значительное количество аккаунтов, на которых кластеризация выдает значение метрики ARI около нуля, что соответствует случайной кластеризации. На рисунке 4 доля этих аккаунтов выделена красным цветом.

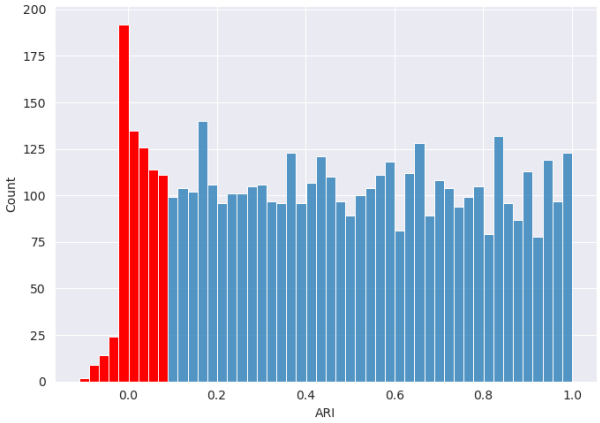

Рис. 4: Распределение значений метрики ARI для отдельных аккаунтов.

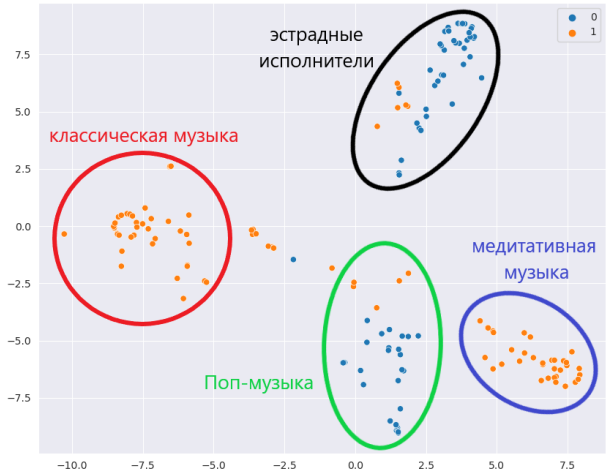

Рис. 5: Координаты точек представляют собой 2 компоненты, полученные применением метода t-SNE к векторным представлениям сессий. Изображение иллюстрирует возможность наличия нескольких паттернов потребления у пользователей.

При подробном анализе нескольких таких аккаунтов, была обнаружена приницпиальная проблема, которая будет далее продемонстрирована на примере совместного аккаунта из синтетического датасета.

На рисунке 5 изображена визуализация сессий одного совместного аккаунта, полученная при помощи алгоритма t-SNE. Точками изображены сессии в совместном аккаунте, выделенные синим и оранжевым в зависимости от реального пользователя. В данном случае синий пользователь слушает поп-музыку и эстрадных исполнителей, а оранжевый пользователь слушает классическую и медитативную музыку.

При этом на рисунке 6 показано, что применение кластеризации объединяет эти жанровые группы таким образом, что метрика ARI и другие метрики получаются близки к случайному угадыванию.

Таким образом, выявлена принципиальная проблема, что если реальные пользователи имеют несколько паттернов потребления, то эти паттерны могут неверно объединяться в один кластер, следствием чего являются очень низкие метрики кластеризации сессий для таких аккаунтов.

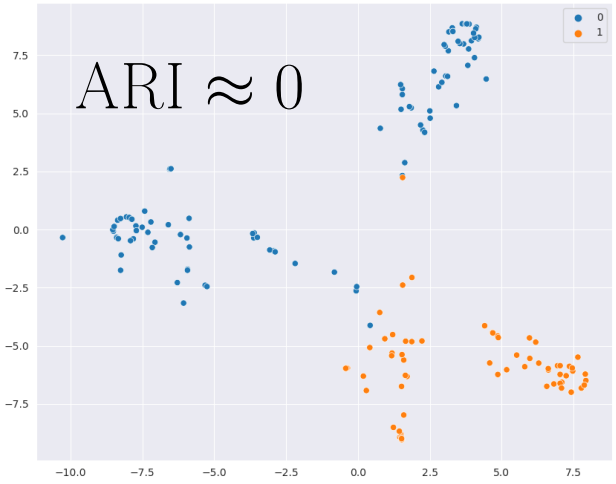

Рис. 6: Координаты точек представляют собой 2 компоненты, полученные применением метода t-SNE к векторным представлениям сессий. Изображение иллюстрирует возможность ошибочной кластеризации сессий совместного аккаунта из-за наличия нескольких паттернов потребления.

Кроме того, эта проблема может быть связана с несовершенством разметки. При использовании синтетических совместных аккаунтов нет никакой гарантии, что каждым из аккаунтов, которые объединяются, пользуется ровно один человек. Аналогичная проблема существует и при использовании разметки по голосу. Кажется распространенной ситуация, когда «взрослому голосу» соответствует несколько реальных пользователей, так как зачастую у детей 2 родителя. Также в семье может быть несколько детей, которые в используемой разметке идентифицируются как «детский голос».

4 Заключение

В ходе исследования задачи кластеризации сессий пользовательской активности в совместных аккаунтах была представлена общая модель для решения поставленной задачи. Были предложены возможные варианты реализации представленной модели с описанием модификаций, которые позволяют улучшить качество работы. На основе

данных музыкального сервиса в голосовом помощнике «Маруся» были сформированы два набора данных, на которых было произведено экспериментальное сравнение предложенных вариантов модели. Отличительной чертой данного исследования стало использование реальной разметки, полученной при помощи распознавания голоса. Также во время проведения экспериментов была обнаружена и в дальнейшем проанализирована принципиальная проблема, связанная с наличием у пользователей нескольких паттернов потребления, что негативно сказывается на кластеризации таких совместных аккаунтов.

В рамках дальнейших исследований по данной теме хорошим направлением кажется кластеризация сессий не на уровне отдельных реальных пользователей, а на уровне паттернов потребления. При такой постановке задачи фактически не требуется разметка, так как можно использовать внутренние метрики качества кластеризации. Также в настоящей работе не поднимался вопрос определения количества кластеров, в проведенных экспериментах число кластеров полагалось известным.

На защиту выносится:

- Описание общей модели для кластеризации сессий пользовательской активности.
- Реализация конкретных вариантов представленной общей модели с описанием модификаций, которые позволяют улучшить качество работы.
- Подготовка двух наборов данных для оценки качества представленных подходов.
- Проведение вычислительных экспериментов на собранных наборах данных для сравнения представленных подходов.
- Анализ проблемы, связанной с наличием у пользователей нескольких паттернов потребления.

5 Список литературы

[1] *CAMRa '11: Proceedings of the 2nd Challenge on Context-Aware Movie Recommendation*, New York, NY, USA, 2011. Association for Computing Machinery.

[2] Enrique Amigó, Julio Gonzalo, Javier Artiles, and M. Verdejo. Amigó e, gonzalo j, artiles j et ala comparison of extrinsic clustering evaluation metrics based on formal constraints. inform retriev 12:461-486. *Information Retrieval*, 12:461–486, 10 2009.

[3] Payal Bajaj and Sumit Shekhar. Experience individualization on online tv platforms through persona-based account decomposition. In *Proceedings of the 24th ACM International Conference on Multimedia*, MM '16, page 252–256, New York, NY, USA, 2016. Association for Computing Machinery.

[4] Rui Chen, Qingyi Hua, Yan-Shuo Chang, Bo Wang, Lei Zhang, and Xiangjie Kong. A survey of collaborative filtering-based recommender systems: From traditional methods to hybrid methods based on social networks. *IEEE Access*, 6:64301–64320, 2018.

[5] Brendan J. Frey and Delbert Dueck. Clustering by passing messages between data points. *Science*, 315(5814):972–976, 2007.

[6] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE International Conference on Data Mining*, pages 263–272, 2008.

[7] Jyun-Yu Jiang, Cheng te Li, Yian Chen, and Wei Wang. Identifying users behind shared accounts in online streaming services. *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 2018.

[8] Yehuda Koren and Robert Bell. *Advances in Collaborative Filtering*, pages 77–118. Springer US, Boston, MA, 2015.

[9] Muyang Ma, Pengjie Ren, Yujie Lin, Zhumin Chen, Jun Ma, and Maarten de Rijke. π-net: A parallel information-sharing network for shared-account cross-domain sequential recommendations. In *Proceedings of the 42nd International ACM SIGIR*

Conference on Research and Development in Information Retrieval, SIGIR'19, page 685–694, New York, NY, USA, 2019. Association for Computing Machinery.

[10] Daniel Müllner. Modern hierarchical, agglomerative clustering algorithms, 2011.

[11] Yuanzhe Peng. A survey on modern recommendation system based on big data, 2022.

[12] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback, 2012.

[13] Gábor Takács and Domonkos Tikk. Alternating least squares for personalized ranking. In *Proceedings of the Sixth ACM Conference on Recommender Systems*, RecSys '12, page 83–90, New York, NY, USA, 2012. Association for Computing Machinery.

[14] Koen Verstrepen and Bart Goethals. Top-n recommendation for shared accounts. In *Proceedings of the 9th ACM Conference on Recommender Systems*, RecSys '15, page 59–66, New York, NY, USA, 2015. Association for Computing Machinery.

[15] Shoujin Wang, Longbing Cao, Yan Wang, Quan Z. Sheng, Mehmet Orgun, and Defu Lian. A survey on session-based recommender systems, 2019.

[16] Zhijin Wang, Yan Yang, Liang He, and Junzhong Gu. User identification within a shared account: Improving ip-tv recommender performance. In *Symposium on Advances in Databases and Information Systems*, 2014.

[17] Markus Weimer, Alexandros Karatzoglou, and Alex Smola. Improving maximum margin matrix factorization. volume 72, 09 2008.

[18] Xinyu Wen, Zhaohui Peng, Shanshan Huang, Senzhang Wang, and Philip S. Yu. Miss: A multi-user identification network for shared-account session-aware recommendation. In Christian S. Jensen, Ee-Peng Lim, De-Nian Yang, Wang-Chien Lee, Vincent S. Tseng, Vana Kalogeraki, Jen-Wei Huang, and Chih-Ya Shen, editors, *Database Systems for Advanced Applications*, pages 228–243, Cham, 2021. Springer International Publishing.

[19] Shuo Yang, Somdeb Sarkhel, Saayan Mitra, and Viswanathan Swaminathan. Personalized video recommendations for shared accounts. *2017 IEEE International Symposium on Multimedia (ISM)*, pages 256–259, 2017.

[20] Amy Zhang, Nadia Fawaz, Stratis Ioannidis, and Andrea Montanari. Guess who rated this movie: Identifying users through subspace clustering, 2014.

[21] Yafeng Zhao, Jian Cao, and Yudong Tan. Passenger prediction in shared accounts for flight service recommendation. In *IEEE Asia-Pacific Services Computing Conference*, 2016.