ДОКЛАДЫ РОССИЙСКОЙ АКАДЕМИИ НАУК. МАТЕМАТИКА, ИНФОРМАТИКА, ПРОЦЕССЫ УПРАВЛЕНИЯ, 2023, том 514, № 2, с. 395–416

НЕ БЫВАЕТ ДВУХ ОДИНАКОВЫХ ПОЛЬЗОВАТЕЛЕЙ: НЕЙРОСЕТЕВАЯ КЛАСТЕРИЗАЦИЯ НА ОСНОВЕ ПОСЛЕДОВАТЕЛЬНОСТЕЙ СОБЫТИЙ ДЛЯ ГЕНЕРАЦИИ АУДИТОРИЙ

© 2023 г. В. Жужель[1,*], В. Грабарь[1], Н. Каплоухая[4], Р. Ривера-Кастро[2,3,4], Л. Миронова[1], А. Зайцев[1], Е. Бурнаев[1]

Представлено академиком РАН А.И. Аветисяном
Поступило 01.09.2023 г.
После доработки 15.09.2023 г.
Принято к публикации 18.10.2023 г.

Определение нужного пользователя для таргетинга является общей задачей для различных интернет-платформ. Хотя многие системы решают ее, они в значительной степени адаптированы к конкретным особенностям. Из-за этого на практике становится непросто применить данные задачи. Причина в том, что большинство систем предназначены для работы с миллионами активных пользователей и с личной информацией, как в случае с социальными сетями или другими сервисами с высокой виральностью. В литературе мало представлены решения, которые предназначены для обработки данных среднего размера, где единственными доступными данными являются последовательности событий пользователя. Это мотивирует нас представить Look-A-Liker (LAL) как систему глубокой кластеризации. Он использует временные точечные процессы для идентификации похожих пользователей для решения задач таргетинга. Для экспериментов мы используем данные ведущего интернет-маркетплейса гастрономического сектора. LAL обобщает не только закрытые данные. Используя последовательности событий пользователей, можно получить результаты мирового уровня, сравнимые с результатами, получаемыми с использованием новых методов, таких как трансформеры и мультимодальное обучение. Наш подход позволяет повысить оценку по метрике ROC AUC до 20% на реальных наборах данных с 0.803 до 0.959. Хотя LAL фокусируется на сотнях тысяч последовательностей, мы показываем, что его можно применить и в задачах с миллионами пользовательских последовательностей. Мы предоставляем полностью воспроизводимую реализацию с кодом и наборами данных в https://github.com/adasegroup/sequence_clusterers.

Ключевые слова: приложения, кластеризация, неконтролируемое обучение, временные точечные процессы

DOI: 10.31857/S2686954323601859, **EDN:** CWIIDI

1. ВВЕДЕНИЕ

С развитием онлайн-рекламы маркетологи могут определять свою целевую аудиторию, используя самые разнообразные подходы. Обычно они указывают такие атрибуты, как демографические группы [37, 39], информация о людях в социальных сетях [21] или граф взаимодействий [54].

Эти подходы, основанные на правилах, извлекают выгоду из опыта маркетологов, но не используют уникальные знания платформы о ее пользователях. Более того, многие платформы, описанные в литературе, как правило, разрабатываются для решения конкретных проблем. Как правило, разработчики систем разрабатывают их с учетом высокого уровня загрузки, подробных профилей пользователей с исчерпывающими описаниями, значительного уровня активности на платформе или характеристик, подобных социальной сети. Мы изучили более двадцати систем расширения аудитории, представленные за последние десять лет. Беглый просмотр литературы показывает, что многие обсуждаемые системы трудно экстраполировать на другие контексты. Воспроизводимость затруднена из-за отсутствия общедоступных реализаций, кроме того, авторы часто экспериментировали только с проприетарными наборами данных.

В качестве примера мы рассмотрим ведущий маркетплейс для гастрономии – Choco. На его

[1]*Сколковский институт науки и технологий, 121205, Москва, Россия*
[2]*Choco Communications, Hasenheide 54, 10967 Берлин, Германия*
[3]*Центр цифровых технологий и управления, Arcisstr. 21 80333 Мюнхен, Германия*
[4]*Исследования, проведенные в период работы в Сколковском институте науки и техники, 121205, Москва, Россия*
*****E-mail: vladislav.zhuzhel@skoltech.ru*

платформе насчитывается более 15 000 ресторанов и 10 000 поставщиков. Мы заинтересованы в предоставлении современного решения, ориентированного на аудиторию, для оптовиков, заинтересованных в таргетинге на рестораны, демонстрирующие специфическое поведение. По сравнению с другими интернет-ресурсами, платформа не собирает характеристики ресторанов в своей системе, и в ней отсутствуют характеристики социальных сетей, такие как графы пользовательской активности. Кроме того, операторы ресторанов демонстрируют различные модели использования. Они делают заказы время от времени и время от времени открывают приложение. В результате ему не хватает свойств популярных социальных сетей, что приводит к необходимости поиска новых решений для расширения аудитории.

Большинство систем расширения аудитории предусматривают два этапа. На первом этапе система нацелена на создание пула пользователей на основе их характеристик, а на втором этапе у них есть модель, которая использует этот пул, например, для отбора пользователей для маркетинговой кампании. Мы ограничиваемся тестированием нашей системы на первом этапе.

Более того, мы ожидаем, что предлагаемый подход будет работать с небольшим объемом размеченных данных и быстро адаптируется к использованию новых продуктов.

Исходя из этих соображений была разработана архитектура Look-A-Liker (LAL). LAL – это архитектура нейронной сети, которая использует вероятностную природу временного точечного процесса, чтобы естественным образом встроить его в алгоритм максимизации математического ожидания [4]. Нейронная сеть LAL предсказывает функцию интенсивности, отвечающую за вероятность событий различных типов. Нейронная сеть является хорошим выбором для моделирования функции интенсивности, поскольку она позволяет охватить нелинейные зависимости. Кроме того, для обучения не требуются размеченные классы в данных. Чтобы обучить модель, мы оптимизируем интенсивность с помощью EM-алгоритма [4].

В прикладной задаче показано, что эта процедура является достаточно общей для проведения второго этапа расширения аудитории.

Наша работа направлена на решение следующих основных задач:

• Встраивание вероятностей временных точечных процессов в EM-алгоритм.

• Улучшение стабильности обучения нейронной сети для исключения вырожденных решений.

• Масштабирование решения для больших наборов данных с миллионами точек без потери качества.

• Поиск способов применения предложенного алгоритма к реальным задачам.

• Достижение оптимального качества прогнозирования без применения других данных, таких как пол, адрес, структура графа и т.д.

Основной вклад этой работы заключается в следующем:

• Разработана модель подобия пользователей, которая **обучается без учителя**. Модель не требует меток для принятия решения, и менее уязвима к изменениям в распределении меток набора данных

• **Модель не требует характеристик пользователя**, таких как пол, адрес и т.д. Таким образом, мы можем избежать хранения огромных объемов персональных данных и предотвратить искажение данных.

• Работает в **режиме холодного старта**. База данных требует лишь ограниченных знаний о клиенте.

• Работает с **наборами данных среднего размера**, что более полезно для большинства интернет-компаний, которые работают с сотнями тысяч пользователей, а не с миллионами, как описано в литературе.

• **Модель имеет широкую применимость** и может использоваться для решения различных прикладных задач, таких как персонализация, управление взаимоотношениями с клиентами или продажи, что отличает ее от большинства узконаправленных частных решений.

• Подход является сравнительно простым и основан на EM-алгоритме, который опирается на парадигму временных точечных процессов.

• **Модель конкурентоспособна на мировом уровне**, достигая результатов, сравнимых с другими современными моделями, такими как архитектура трансформеров.

2. ОБЗОР ЛИТЕРАТУРЫ

Кластеризация последовательностей событий лежит в основе задачи по отбору групп похожих пользователей. Чтобы правильно решить эту задачу, нужно обратить внимание на особенности имеющихся данных, поскольку модели для последовательностей событий отличаются от моделей для широкого применения и требуют особого подхода к обучению и логическому выводу.

Ниже мы начнем с обзора поведенческого таргетинга с помощью дополнительной модели кластеризации и без нее. Затем мы переходим к моделям последовательностей событий как реализациям точечных процессов и к тому, как должна выполняться кластеризация в этом случае.

Контролируемый поведенческий таргетинг. При контролируемом обучении мы считаем, что у

нас есть метки, которые мы просим нашу модель предсказать. В этом случае можно использовать классические подходы к машинному обучению.

Работа [33] нацелена на поведенческий таргетинг путем максимизации конверсий. В основе подхода лежит история посещений пользователя. [27] предлагает систему онлайн-рекламы, основанную на истории пользователей. Авторам требуются конверсии пользователей в качестве меток. [56] включает представления на уровне фраз для системы аннотаций. Авторы фокусируются на текстовых моделях и обучении с учителем. [55] рассматривает двухэтапный подход. Авторы используют графовую нейронную сеть для оффлайн-этапа. Для онлайн-этапа авторы применяют дистилляцию знаний. [28] предлагает скоринговую модель для моделирования двойников. [5] изучает представления и представляет модель оценки данных социальных сетей для расширения аудитории. Работа [10] также рассматривает применение современных нейронных архитектур для создания последовательностей событий для контролируемого сценария. Авторы подчеркивают, что нужны нейронные сети для наилучшей работы в контролируемом сценарии. Еще одной выгодной стратегией повышения точности таргетинга в электронной коммерции является включение контекстной информации. Например, модели покупок потребителей могут варьироваться в зависимости от того, покупают ли они для себя или дарят другим. Полезность такой контекстуально насыщенной информации была показана в [8]. Подход, основанный на сеансах, при котором анализ фокусируется на отдельных сеансах пользователя на платформе, а не на их полной исторической последовательности, является многообещающим методом интеграции этого контекста в системы электронной коммерции [20]. Используя архитектуры, основанные на рекуррентных нейронных сетях (RNN), авторы продемонстрировали, что системы рекомендаций могут быть значительно улучшены с помощью данных потока кликов на основе сеанса.

Во многих работах рассматривается усовершенствование модели “на лету” с использованием онлайн-обучения [35]. В частности, [55] применяет дистилляцию знаний для онлайн-обучения. [18] следует подходу извлечения признаков в сочетании с прогнозированием профиля. Авторы используют онлайн-обучение для решения проблемы расширения аудитории. [7] изучает методы поиска новых и релевантных пользователей в рекламной кампании путем расширения существующего, меньшего набора пользователей. Авторы предлагают новый вид аналогичной модели на основе автокодирования, в которой используется состязательное обучение, чтобы систематически “поощрять” промежуточный слой быть бинарным. [53] объединяет онлайн и оффлайн стадии. Они предоставили реализацию, но набор данных трудно получить. Внимание авторов сосредоточено на больших наборах данных (больше 200 тысяч последовательностей с очень богатыми данными). Для этого они предоставляют обширное сравнение. Другая идея авторов заключается в том, чтобы использовать варианты обучения подобию, поискав ближайших соседей в пространстве исходных признаков или представлений, полученных на выходе нейронной сети. Входные данные могут различаться: [22] представляет систему, основанную на данных социальных сетей, [25] доказывает, что этот подход работает с графовыми данными (в качестве входных данных и последующего поиска ближайших соседей), [13] реализует чувствительное к локальности хеширование для повышения эффективности использования времени и памяти в подходах, основанных на обучении по сходству.

Неконтролируемый поведенческий таргетинг. Если нет размеченных данных, нужно искать другие пути. Ниже мы приводим примеры различных методов и архитектур, используемых в неконтролируемом обучении. [23] использует механизм привлечения внимания. Здесь они объединяют слой внимания вместо слоя объединения. Авторы используют это для изучения действий пользователей в режиме реального времени без контроля. [36] представляет систему для онлайн-рекламы, использующую графические данные неконтролируемым образом. [2] предоставляет механизм расширения аудитории для моделирования двойников. [24] – это двухступенчатая система расширения аудитории, включающая в себя смещенную обратную связь с семенами в мета-обучении для поиска золотых семян из зашумленного набора семян. [44] использует историю просмотров в последовательном порядке. Задача состоит в том, чтобы провести обучение представлений без контроля, где каждый пользователь представляет собой набор URL-адресов и моделируется как один документ. В другой работе [41] доказано, что успешное обнаружение аномалий может быть достигнуто в неконтролируемой среде. [26] использует хэширование с локальной чувствительностью. Они рассматривают таргетинг как проблему рекомендательной системы и проверяют свой подход, используя наборы данных, популярные в литературе по рекомендательным системам. Наборы данных велики, но в них отсутствуют временные метки, и они используют данные только за один день. [34] представляет собой метод трансферного обучения. Они фокусируются на пользователях Интернета и используют данные об их URL-адресах.

Наконец, [39] представляет систему, состоящую из логических правил.

Более общий подход заключается в использовании кластеризации аудиторий: необходимо построить модель, которая не использует размеченные данные, а вместо этого группирует пользователей в кластеры. Таким образом, модель присваивает вероятности принадлежности к каждому кластеру для каждого пользователя. Таким образом, впоследствии таргетинг становится простым: мы выбираем наиболее релевантный кластер и ориентируемся на пользователей из него.

[48] посвящена задаче кластеризации. В работе объединяются графы и последовательности пользователей. Авторы статьи подробно описывают реализацию, но не публикуют данные. Эта работа хорошо подходит для очень плотных наборов данных с онлайн-активностями. Авторы нормализуют пользовательские последовательности с помощью сигмоидных функций. Эта нормализация приводит к признакам, которые позже используются в качестве входных данных при кластеризации методом K-means на следующем этапе.

В новом подходе [50] для эффективной кластеризации используется динамическое расстояние искривления времени. Эксперименты подтверждают, что такой подход хорошо подходит для работы с неоднородностью, присущей таким данным.

Последовательности событий как временные точечные процессы. Как мы уже говорили, вполне естественно изучать пользователей, рассматривая их последовательности событий. Это быстро развивающаяся область с многочисленными приложениями моделей глубокого обучения [38, 47]. Помимо временной структуры, можно рассматривать и альтернативные структуры, например графовые. Примером такого подхода является предсказание информационных каскадов на основе этих структур, что рассматривается в [45].

Мы хотим отдельно обсудить основные достижения в этой области, которые имеют отношение к временным точечным процессам Хоукса [14].

Мощность рекуррентных нейронных сетей (РНС) является общепризнанным фактом. В исследовании [29] показано, что они обеспечивают производительность, превосходящую классические подходы. Дальнейшее подтверждение эффективности РНС можно найти в [30], где успешно решена реальная задача моделирования распространения информации в социальных сетях во время пандемии COVID-19. В последующих работах рассматривались приложения одномерных сверточных нейронных сетей [40] и трансформаторов к проблеме моделирования последовательностей событий [57]. В этих работах выделены два требования к моделям последовательностей событий: дальнодействующая память и эффективность. Трансформеры страдают от неэффективности для длинных последовательностей [42, 43], а сверточные нейронные сети имеют ограниченную память [32]. Таким образом, для проверки гипотезы об эффективной кластеризации последовательностей событий мы можем ориентироваться на рекуррентные нейронные сети.

Кластеризация нейронных временных точечных процессов. Идея кластеризации была недавно принята в некоторых статьях.

В статье [46] представлен тип EM-алгоритма для работы с нейросетевыми моделями временных точечных процессов. Юньхао Чжан в статье [51] предложил подход, основанный на нейронных временных процессах. Авторы показали превосходство нейронных сетей в кластеризации последовательностей событий и возможность применения подходов типа EM. В обеих работах отмечается сложность данной задачи, поскольку сходимость неустойчива и требует тщательной настройки гиперпараметров. Отметим, что эти вопросы носят более общий характер: для самоподдерживающегося обучения можно определить функцию потерь, основанную на кластеризации [1, 17], в то время как альтернативные подходы, не основанные на кластеризации, показывают более высокие метрики производительности [15].

Итог. Мы видим, что существует мало литературы, связанной с кластеризацией последовательностей событий. Основная задача здесь заключается в том, что нам нужен подход глубокого обучения для эффективной обработки данных о последовательности событий – и в этом случае трудно обеспечить стабильный и эффективный подход к кластеризации. Мы хотели бы восполнить этот пробел, предложив эффективный и стабильный подход к глубокому обучению для кластеризации последовательности событий. Такой подход решает проблему кластеризации пользователей, представленных их последовательностями событий, и нацеливания на каждый кластер отдельно.

3. ПРЕДВАРИТЕЛЬНЫЕ СВЕДЕНИЯ

Последовательность событий $\mathbf{s}$ – это последовательность пар $\{(t_i, c_i)\}$, где t_i – временная метка наступления события, а c_i $\{1, 2, \ldots, C\}$ – тип события.

Наша выборка состоит из N примеров последовательностей событий $\mathbf{S} = \{\mathbf{s}_n\}_{n=1}^{N}$. Для каждого события в последовательности мы знаем время события и тип события.

Последовательности событий являются частным случаем реализаций временного точечного процесса. Если $|C| > 1$, то процесс называется процессом с отмеченной временной точкой.

Одним из широко используемых определений для описания временных точечных процессов является функция интенсивности [3].

Для каждого типа события c можно определить свою условную функцию интенсивности $\lambda_c(t|\mathcal{H}_t)$, где $\mathcal{H}_t = \{(t_i, c_i)|t_i < t\}$ – история до времени t. Функция интенсивности имеет вид:

$$\lambda_c(t) = \frac{\mathbb{E}[dN_c(t)|\mathcal{H}_t]}{dt}, \quad (1)$$

где $N_c(t)$ – счетная функция, т.е. количество событий типа c, появившихся до момента t, или более формально как

$$N_c(t|\mathcal{H}_t) = \#\{(t_i, c_i)|t_i < t, c_i = c\}. \quad (2)$$

Мы можем записать и оптимизировать правдоподобие данных в терминах функции интенсивности.

Процесс Пуассона. Пуассоновский процесс является простейшей моделью, рассматриваемой с использованием функции интенсивности. В ней предполагается, что события независимы. Для однородного пуассоновского процесса мы предполагаем постоянную функцию интенсивности, $\lambda_c(t) = \lambda_c$. У неоднородных пуассоновских процессов интенсивность зависит только от времени, $\lambda_c(t)$. Определим ее как

$$\forall(t_i, c_i), t : p(t_i \in [t; t + dt)|t_i > t, c = c_i) = \lambda_{c_i}(t). \quad (3)$$

В машинном обучении мы используем параметрическое предположение для функции интенсивности.

Процесс Хоукса. Для учета истории событий можно использовать процесс Хоукса [14].

Это самовозбуждающийся процесс с аддитивным воздействием.

Это означает, что события положительно влияют на интенсивность в будущем. Функция интенсивности для процесса Хоукса имеет вид

$$\forall c \in \mathcal{C} : \lambda_c(t) = \mu_c(t) + \sum_{i:t_i<t} \varphi_{cc_i}(t - t_i). \quad (4)$$

Здесь $\mu_c(t)$ – базовая интенсивность, аналогичная пуассоновскому процессу. $\varphi_{cc'}(s)$ – так называемая функция воздействия, учитывающая предыдущие события.

В случае процесса Хоукса мы предполагаем, что $\mu_c(t) \geq 0$ и $\varphi_{cc'}(s) \geq 0$. Таким образом, основными ограничениями для точечных процессов Хоукса являются аддитивность и неотрицательность воздействий.

Эти ограничения лимитируют реальные процессы, моделируемые с помощью точечного процесса Хоукса.

3.1. Алгоритм EM

Как можно заметить, временные точечные процессы могут быть описаны вероятностным образом. Это свойство позволяет нам естественным образом использовать EM-алгоритм с моделями, основанными на интенсивности.

Целевой функцией EM-алгоритмов является математическое ожидание логарифмической вероятности. Этот метод полностью описывается в терминах вероятностей. Таким образом, в него можно включить вероятности временных точечных процессов, поскольку закон временных точечных процессов является вероятностным законом, как показано в предыдущем разделе. Тот факт, что мы можем использовать EM-алгоритм без дополнительных предположений о природе данных (например, гауссово распределение данных, с которым можно столкнуться в GMM), за исключением того факта, что существует некоторая функция интенсивности, вдохновил нас на использование этой комбинации.

4. МЕТОДЫ

Предлагаемый нами алгоритм нейронной кластеризации последовательностей, моделируемых как временные точечные процессы LAL, состоит из нескольких компонентов. К ним относятся:

1. *Разбиение*. Использование разбиения на равные интервалы для подготовки входных данных для нейронной сети.

2. *Кодирование представления*. Обработка категориальных разделов для получения представлений с помощью LSTM.

3. *Кластеризация*. Модель поверх полученных представлений.

Мы обобщаем работу конвейера в рис. 1. Технические подробности приведены ниже.

Мы также описываем нашу стратегию повышения стабильности процесса, поскольку стандартный подход подвержен графики пользователей нестабильности.

4.1. Разбиение данных точечного процесса

Предположим, что интенсивность функции для последовательности задана как $\lambda_c(t|\mathcal{H}_t) \in L_1[0, T_s]$, где T_s – конечное время последовательности. Мы можем рассмотреть разбиение на части в качестве приближения для данной интенсивности. В этом случае вместо обработки последовательностей мы обрабатываем разбиения.

Для заданной последовательности из набора датасетов $\mathbf{s} \in \mathbf{S}$ и конечного времени T_s для этой последовательности мы разделяем интервал на n_{steps} подынтервалов размером $\Delta t = T_s/n_{steps}$.

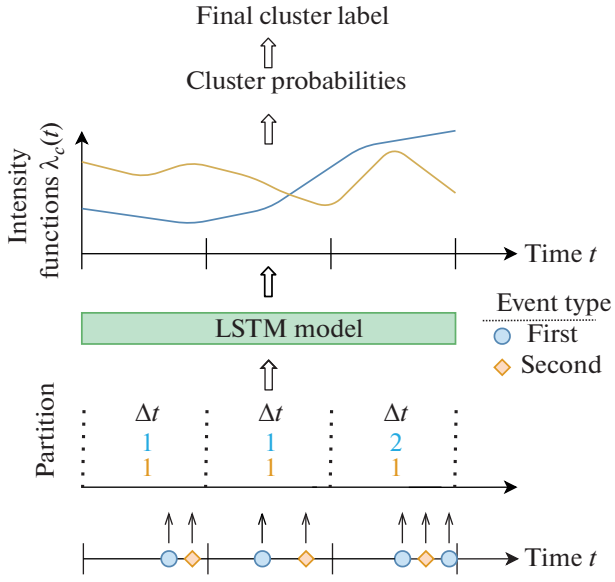

Рис. 1. Обзор системы LAL для нейронной кластеризации маркированных временных точечных процессов.

Подсчитывая количество событий каждого типа в каждом интервале, получаем матрицу разбиения $\mathbf{P} \in (\mathbb{N} \cup \{0\})^{C \times n_{steps}}$. Затем мы добавляем строку со значением Δt. Таким образом, разбиение является функцией, которая отображает последовательность событий в матрицу.

$$P(\mathbf{s}, n_{steps}) : \mathbf{S} \times \mathbb{N} \to \mathbb{R}^{(C+1) \times n_{steps}}. \quad (5)$$

Представим каждый элемент матрицы разбиения $\mathbf{P} = \{P_{ij}\}$, как

$$P_{ij} = \#\{s_k = (c_k, t_k) \in \mathbf{s} | c_k = i, t_k \in [j\Delta t, (j+1)\Delta t)\}, \quad (6)$$

где $i \neq 0$ равно количеству событий типа i во временном интервале $[j\Delta t, (j+1)\Delta t)$. Для возможности выбора различных Δt мы включаем их в качестве нулевой строки матрицы разбиения, $P_{0j} = \Delta t$, $j = \overline{1, n_{steps}}$.

После обработки всех последовательностей событий необходимо нормировать значения Δt в матрице P, например, так

$$\Delta t \leftarrow \frac{\Delta t - \min(\Delta T)}{\max(\Delta T) - \min(\Delta T)}. \quad (7)$$

Здесь $\Delta T = \{\Delta t_i\}_{i=1}^{N}$. Такая нормировка позволяет достичь численной устойчивости для произвольного числа n_{steps}.

Выбор n_{steps} . Наши эксперименты показывают, что число шагов должно быть несколько больше средней длины последовательности. Если n_{steps} мало, то разрешение недостаточно хорошо для улавливания значимых зависимостей, а если n_{steps} слишком много, то модели гораздо сложнее уловить долгосрочные зависимости. Мы используем $n_{steps} = 128$ для всех синтетических наборов данных и корректируем это значение для реальных наборов данных. Мы рекомендуем использовать $n_{steps} \approx 2N_{avg}$, где N_{avg} – средняя длина последовательности. Однако окончательное решение о величине n_{steps} остается за пользователем и может зависеть от набора данных.

4.2. Кодировка представления с помощью модели LSTM

Мы рассматриваем LSTM для обработки разбиения и прогнозирования интенсивности. LSTM – это нейросетевая архитектура, предназначенная для моделирования последовательных данных [11]. LSTM вычисляет последовательность, используя один элемент за раз. Блок обработки одинаков для всех элементов. Вход блока для t-го элемента последовательности состоит из новой информации, доступной на t-м шаге, $\mathbf{x}_t$ из $\mathbf{s}$, а также скрытых состояний и состояний ячейки, $\mathbf{h}_{t-1}, \mathbf{c}_{t-1} \mathbb{R}^D$ с памятью о прошлых событиях. Выходом блока LSTM, $\mathbf{h}_t, \mathbf{c}_t = g(\mathbf{x}_t, \mathbf{h}_{t-1}, \mathbf{c}_{t-1})$, являются новые скрытые состояния и состояния ячейки. Клеточное состояние позволяет улавливать долгосрочные зависимости. Скрытое состояние – это текущее представление прошлого, ориентированное на краткосрочные зависимости. После блока LSTM мы используем одномерную пакетную нормализацию [16].

Мы начинаем с предположения, что функции интенсивности для каждого типа событий λ_c постоянны в момент времени, и оцениваем их с помощью LSTM. Аналогично [29], мы записываем функцию интенсивности в виде

$$\lambda_c(t) = f_c(\mathbf{w}_c^\top \mathbf{h}_t), \quad (8)$$

где функция активации имеет вид

$$f_c(x) = b_c \log(1 + \exp(x/b_c)). \quad (9)$$

Здесь b_c и $\mathbf{w}_c$ являются обучаемыми параметрами.

Для восстановления матрицы интенсивностей $\boldsymbol{\lambda} \in \mathbb{R}_{0+}^{C \times n_{steps}}$ для последовательности $\mathbf{s}$, мы пошагово применяем LSTM-модель, представленную в виде $\boldsymbol{\lambda} = LSTM(P(\mathbf{s}, n_{steps}), \mathbf{h}_0)$. Следовательно, log-вероятность разбиения аналогична [29] и $\log p(P(\mathbf{s}, n_{steps}) | \boldsymbol{\lambda})$ становится

$$\sum_{i,j} -\lambda_{i,j}\Delta t + P_{i+1,j} \log(\lambda_{i,j}\Delta t) - \log(P_{i+1,j}!). \quad (10)$$

Для такого временного точечного процесса мы максимизируем логарифмическую вероятность относительно параметров Θ_{LSTM}, состоящих из параметров блока LSTM и параметров $\mathbf{w}_c$ и b_c для типов событий $c \in \{1,\ldots,C\}$.

4.3. Кластеризация на основе модели LSTM

Мы рассматриваем K кластеров и N реализаций $\{\mathbf{s}_n\}_{n=1}^N$ последовательностей событий. Используя эти обучающие последовательности, мы хотим найти кластеры среди них и различные функции интенсивности для разных кластеров.

Для k-го кластера скрытые состояния $\mathbf{h}_{ik}$ изменяются в процессе обработки последовательности событий с начальными точками, соответствующими начальному скрытому состоянию кластера $\mathbf{h}_{0k}$. Аналогичным образом мы можем обучить соответствующие состояния ячейки. Таким образом, у нас есть K начальных скрытых состояний, и мы представляем их как $\mathbf{H} = \{\mathbf{h}_{0k}\}_{k=1}^K$. Стоит отметить, что это фундаментальное отличие от подобного подхода в [51], потому что вместо обучения нескольких нейронных сетей мы обучаем только одну и различаем кластеры только по начальным скрытым состояниям. Этот факт увеличивает масштабируемость модели.

Мы ожидаем, что наша модель может предсказывать интенсивностную функцию $\boldsymbol{\lambda}_k = LSTM(P(\mathbf{s}, n_{steps}), \mathbf{h}_{0k})$, и параметры модели $LSTM$ остаются неизменными.

Теперь рассмотрим смесевую модель с вероятностями смешивания для кластеров $\pi_k \geq 0$ и $\sum_{k=1}^K \pi_k = 1$.

Для простоты определим $P_n = P(\mathbf{s}_n, n_{steps})$ и $\boldsymbol{\lambda}^k(P_n)$ равным

$$p(P_n|\mathbf{h}_{0k}) = p(P_n|\boldsymbol{\lambda}^k(P_n)) = \prod_{i,j} \frac{e^{-\lambda_{i,j}^k \Delta t}(\lambda_{i,j}^k \Delta t)^{P_{i+1,j}}}{P_{i+1,j}!}. \quad (11)$$

Здесь $\lambda_{i,j}$ является интенсивностью i-th события на j-th временной метке.

Обозначим $\mathbf{z}$ как латентную переменную, соответствующую конкретной последовательности $\mathbf{s}$. Следовательно, $z_k = 1$, если $\mathbf{s}$ принадлежит к k-th кластеру и нулю в противном случае. Для всего набора, мы имеем матрицу латентных переменных, $\mathbf{Z} = (\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_N)$. Поэтому мы можем записать $\pi_k = p(z_k = 1)$ и $p(\mathbf{z}) = \prod_{k=1}^K \pi_k^{z_k}$.

Таким образом, мы видим, что

$$p(P_n|\mathbf{z}) = \prod_{k=1}^K p(P_n|\mathbf{h}_{0k})^{z_{nk}}, \quad (12)$$

где K – это число кластеров в модели, и что вероятность того, что последовательность $\mathbf{s}$ – это

$$p(\mathbf{s}_n) = \sum_{\mathbf{z}} p(P_n|\mathbf{z})p(\mathbf{z}) = \sum_{k=1}^K \pi_k p(P_n|\mathbf{h}_{0k}). \quad (13)$$

Обозначая множество всех разбиений как $\mathbf{P} = \{P_n\}_{n=1}^N$, мы можем найти вероятность всего набора данных как

$$p(\mathbf{P}) = \prod_{n=1}^N p(P_n). \quad (14)$$

Аналогично, мы имеем

$$\log p(\mathbf{P}) = \sum_{n=1}^N \log \sum_{k=1}^K \pi_k p(P_n \mid \mathbf{h}_{0k}). \quad (15)$$

Для нахождения вероятности кластерной метки мы вычисляем

$$p(z_k = 1|P_n) = \frac{\pi_k p(P_n|\mathbf{h}_{0k})}{\sum_{j=1}^K \pi_j p(P_n|\mathbf{h}_{0j})}. \quad (16)$$

Однако решение прямой задачи неустойчиво. Поэтому мы используем вариант известного EM-алгоритма [4] и снабжаем его всеми полученными к настоящему времени вероятностями.

Целевую функцию EM-алгоритма можно записать следующим образом

$$Q(\Theta^*, \Theta) = \mathbb{E} \log p(\mathbf{P}, \mathbf{Z}|\Theta^*) = \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{P}, \Theta) \log p(\mathbf{P}, \mathbf{Z}|\Theta^*). \quad (17)$$

Эта целевая функция принимает параметры предыдущей итерации Θ и вычисляет ожидание согласно им. Здесь Θ – параметры модели LSTM. Во время тренировки мы должны вычислить новые параметры Θ^* и затем повторить это с новыми параметрами.

Для кластеризации последовательности событий, мы можем переписать это как

$$Q(\Theta^*, \Theta) = \sum_{n=1}^N \sum_{k=1}^K \gamma(z_{nk})[\log \pi_k^* + \log p(P_n|\mathbf{h}_{0k}^*)], \quad (18)$$

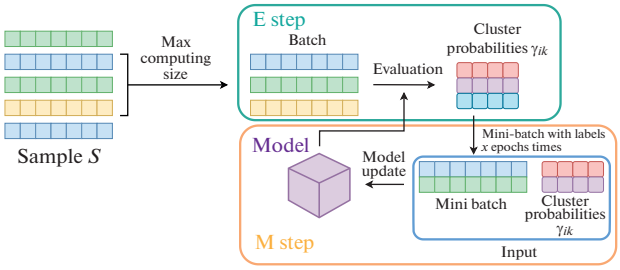

Рис. 2. Одна эпоха EM-алгоритма с процедурой double batching.

где

$$\gamma_{nk} = \gamma(z_{nk}) = p(z_k = 1|P_n) = \frac{\pi_k p(P_n|\mathbf{h}_{0k})}{\sum_{j=1}^{K}\pi_j p(P_n|\mathbf{h}_{0j})},$$

$$\log p(P_n|\mathbf{h_{0k}}) = \tag{19}$$

$$= \sum_{i,j}(-\lambda_{i,j}^k \Delta t + P_{i+1,j}\log(\lambda_{i,j}^k \Delta t) - \log(P_{i+1,j}!)).$$

Во время шага E мы вычисляем γ_{nk} с использованием параметров модели предыдущей итерации. Затем на шаге M мы обновляем параметры модели с помощью нескольких эпох тренировки нейронной сети. В нашем случае мы используем 10 эпох. Мы повторяем эту пару шагов E и M 50 раз.

4.4. Повышение стабильности и эффективности

Процедура двойного батчинга. В ходе работы EM-алгоритма мы оптимизируем следующую целевую функцию:

$$Q(\Theta^*, \Theta) = \mathbb{E}\log p(\mathbf{P}, \mathbf{Z}|\Theta^*) =$$

$$= \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{P}, \Theta)\log p(\mathbf{P}, \mathbf{Z}|\Theta^*). \tag{20}$$

Здесь Θ – набор старых параметров модели, а Θ^* – набор новых параметров после обновления. На шаге E мы вычисляем матрицу с элементами $\gamma_{nk} = p(z_n = k|P_n)$. Это вычисление требует большого объема памяти и занимает много времени для больших выборок S. Чтобы преодолеть эту проблему, мы предлагаем использовать процедуру двойного батчинга, изображенную на рис. 2.

Процедура использует два размера батчей, n_E и n_M. Первый размер n_E – это максимальный размер вычисления. Он определяет количество точек данных для шага E. Второй размер, n_M, – это размер батча, который мы используем на шаге M при обновлении параметров нейронной сети.

При таком подходе и использовании (19) мы оцениваем и оптимизируем одну и ту же целевую функцию, которую описываем как

$$Q_{\max} = \mathbb{E}\log p(\mathbf{P}_{batch}, Z|\Theta^*),$$

$$\mathbb{E}_{batch}Q_{\max} \sim \frac{n_E}{N}\mathbb{E}\log p(\mathbf{P}, Z|\Theta^*), \tag{21}$$

здесь $Q_{\max}$ – количество, вычисленное для партии размером n_E, а второе ожидание – ожидание по всем возможным партиям.

Вместо оптимизации функции потерь, вычисленной на всем наборе данных, мы предлагаем выбирать подмножество максимального размера на каждом шаге E и оптимизировать его. Таким образом, мы оптимизируем $Q_{\max}$ вместо его ожидания, что эквивалентно исходной проблеме. Однако из-за случайного выбора подмножества в долгосрочной перспективе они будут эквивалентными.

Такой подход позволяет использовать предложенный алгоритм EM для наборов данных произвольного размера. Мы демонстрируем правильность этого подхода в наших экспериментах.

Повышение стабильности кластеризации. Как это часто бывает с моделями глубокого обучения, конкретное решение может меняться при множественных запусках. Некоторые кластеры могут быть практически неотличимыми или становиться пустыми, и алгоритм может создавать вырожденные решения, объединяя несколько фактических кластеров в один. Чтобы предотвратить это, мы адаптируем идеи из [46]. Мы предлагаем зафиксировать вероятности смешивания на $\pi_k = \frac{1}{K}$ и использовать алгоритм Монте-Карло по схеме Марковской цепи (MCMC) для выбора числа кластеров.

Случайное блуждание:

Мы инициализируем модель одним кластером и изменяем количество кластеров в соответствии с алгоритмом MCMC на каждом шаге EM-алгоритма. Кроме того, мы ограничиваем максимальное количество кластеров разумным числом. Например, мы используем 10 в качестве верхней границы. Модель может сделать два хода на каждом шаге. Она может уменьшить или увеличить количество кластеров. Если количество кластеров не равно одному или максимальному количеству кластеров, вероятность изменения направления выбора равна $\frac{1}{2}$.

Увеличение:

Мы увеличиваем количество кластеров, разделяя один из них. Случайным образом выбираем a из бета-распределения, $a \sim Be(1,1)$ [52], и вычисляем скрытые состояния для двух новых кластеров как $\mathbf{h}_i \leftarrow 2a\mathbf{h}_i$, $\mathbf{h}_{K+1} \leftarrow 2(1-a)\mathbf{h}_i$, где $\mathbf{h}_{K+1}$ – новое скрытое состояние.

Уменьшение:

С вероятностью $\frac{1}{2}$ модель либо удаляет один кластер, либо объединяет два. В случае объединения i-го и j-го кластеров, начальные скрытые состояния нового кластера после объединения вычисляются как $\mathbf{h}_i^{new} \leftarrow \frac{\mathbf{h}_i + \mathbf{h}_j}{2}$.

Принятие/отклонение:

После первого шага встает вопрос о том, следует ли сохранить новую модель. Мы можем принять это решение вероятностным образом. Вероятность принятия есть минимум между 1 и отношением правдоподобия, $p(\mathbf{P}|K_{new})/p(\mathbf{P}|K_{old})$.

Фиксированное количество кластеров:

Наилучший выбор для максимизации логарифма правдоподобия – бесконечное разделение и обучение на подмножествах. Чтобы предотвратить такое поведение, мы (1) ограничиваем максимальное количество кластеров, (2) принудительно удаляем дополнительные кластеры в течение последних нескольких эпох, например, 10 из 50 эпох алгоритма Expectation-Maximization (EM). Принудительное удаление не ухудшает общую производительность, измеряемую метрикой чистоты.

Такой подход позволяет модели идентифицировать все существующие кластеры, увеличивая их количество. Он также предотвращает неудовлетворительные решения, когда несколько кластеров обучаются вместе как один кластер. В конечном итоге модель принуждается удалить дополнительные кластеры, которые почти пусты или отличаются незначительно от других, что приводит к более практическому решению. Стоит заметить, что следует избегать выбора очень большого верхнего предела для числа кластеров. Это увеличивает вероятность того, что мы получим кластеры, содержащие небольшое количество точек, и алгоритм не сможет удалить их.

4.5. Технические детали

- Количество шагов для разбиений было выбрано близким к удвоенной средней длине последовательности
- для всех экспериментов мы рассматривали 50 эпох EM-алгоритма с 10 эпохами обучения нейронной сети в течение M-шага.
- Размер подмножества, которое использовалось в течение каждой эпохи EM-алгоритма, составлял 2000
- Начальная скорость обучения была установлена на уровне 0.1
- Мы использовали оптимизатор Adam и уменьшали скорость обучения на плато с допуском 25 и множителем 0.5.
- Изначальное число кластеров было установлено равным 1, верхняя граница числа кластеров составляла либо 10, если истинное число кластеров меньше 10, и 20 в противном случае
- В качестве представления последовательности рассматривалось конечное состояние ячейки

5. ЭКСПЕРИМЕНТЫ

Для обеспечения воспроизводимости мы демонстрируем нашу систему на широко доступных наборах данных научному сообществу. Такой подход также подчеркивает способность LAL к обобщению. Мы проводим эксперименты на тринадцати синтетических и четырех реальных наборах данных.

Сводная информация о наборах данных приведена в табл. 1.

5.1. Синтезированный набор данных

Описание набора данных. Одним из основных преимуществ нашего подхода является возможность его применения к массивным наборам данных с большим количеством последовательностей. Для доказательства этого мы генерируем набор данных с последовательностями общим количеством в один миллион точек и проводим обучение на этих данных с максимальным размером вычислений, равным 2000.

Мы генерируем этот набор данных с помощью пяти процессов Хоукса с экспоненциальным затуханием. Более формально мы определяем его как

$$\lambda_c(t|\mathcal{H}_t) = \mu_c + \sum_{i:t_i<t} a_{cc_i}\delta_{cc_i}\exp(-\delta_{cc_i}(t-t_i)). \qquad (22)$$

Здесь мы предполагаем, что $\mu_c \sim U[0;1]$, $a_{cc'}$, $\delta_{cc'} \sim U([0;0.6])$. Мы используем метрику чистоты для оценки качества кластеризации в (23).

$$\text{Purity}(C,R) = \frac{1}{N}\sum_{k=1}^{K}\max_{j\in\{1,\ldots,K'\}}|C_k \cap R_j|. \qquad (23)$$

Здесь C_k – это множество событий типа k во множестве C set. R_j эквивалентно.

В рис. 3 мы визуализируем синтетические данные. Стоит отметить, что три кластера расположены очень близко друг к другу, и их бывает трудно различить, например, на рис. 3б.

Чтобы продемонстрировать способность модели работать с разнородными данными, мы сгенерировали двенадцать дополнительных синтетических наборов данных с меньшим количеством точек данных. Мы провели эксперименты

Таблица 1. Статистика наборов данных: название набора данных и соответствующая средняя длина последовательности, количество последовательностей в наборе данных, количество типов событий и количество кластеров (если применимо), здесь Amazon 7.5 тыс. является подмножеством Amazon 100 тыс.

Датасет	Средняя Длина последовательности	Nr. of Последовательности	Nr. of Типы событий	Nr. of True Кластеры
Real-world датасеты				
Age	862.44	30000	64	–
IPTV	3225.159	302	16	–
LinkedIn	3.073	2439	82	–
Amazon 7.5 тыс.	56.5	7523	8	–
Amazon 100 тыс.	12.2	102671	8	–
Синтетические данные				
Huge	50	10^6	5	5
K2_C5	50	800	5	2
K3_C5	50	1200	5	3
K4_C5	50	1600	5	4
K5_C5	50	2000	5	5
sin_K2_C5	143.5	800	5	2
sin_K3_C5	186.1	1200	5	3
sin_K4_C5	185.1	1600	5	4
sin_K5_C5	119.3	2000	5	5
trunc_K2_C5	52.9	800	5	2
trunc_K3_C5	57.9	1200	5	3
trunc_K4_C5	53.3	1600	5	4
trunc_K5_C5	53.7	2000	5	5

по сравнению нашей модели с моделью Дирихле-Хоукса [46] и трансформерной моделью Хоукса [57]. Конвейер генерации такой же, как и в [46], но расширенный. Восемь наборов данных были сгенерированы так же, как и в [46], с использованием синусоидальных функций воздействия и кусочно-постоянных функций воздействия, соответственно, а остальные четыре – с помощью моделирования экспоненциального ядра.

Для каждого вида формы функции влияния было сгенерировано четыре набора данных с различным количеством кластеров. Количество кластеров было выбрано равным 2, 3, 4 и 5. Каждый кластер содержит 400 последовательностей событий, а каждая последовательность событий содержит 5 типов событий. Интенсивность экзогенного базиса равномерно дискретизировалась из [0, 1]. Каждая синусоидальная функция воздействия в k-th кластере имеет вид $\phi^k_{cc'} = b^k_{cc'}(1 - \cos(\omega^k_{cc'}(t - s^k_{cc'})))$, где $\{b^k_{cc'}, \omega^k_{cc'}, s^k_{cc'}\}$ выбирается случайным образом из $\left[\frac{\pi}{5}, \frac{2\pi}{5}\right]$.

Каждая кусочно-постоянная функция воздействия является усечением соответствующей синусоидальной функции воздействия, т.е. $2b^k_{cc'} \times \text{round}(\phi^k_{cc'}/(2b^k_{cc'}))$.

Для синтетических наборов данных с синусоидальной функцией влияния используются следующие обозначения: sin-Kn-C5, где n – число кластеров, от 2 до 5. Для наборов данных, сгенерированных с использованием кусочно-постоянной функции воздействия, использовалось название trunc-Kn-C5.

Аналогично, четыре набора данных с экспоненциальным ядром были сгенерированы с именами Kn-C5.

Эксперименты

Масштабируемость

Чтобы проверить, как влияет на производительность максимальный размер вычислений в рис. 2, мы обучаем модель с одним и тем же гиперпараметром и максимальным размером вычислений, но с разными размерами массивов данных. Максимальный объем вычислений во

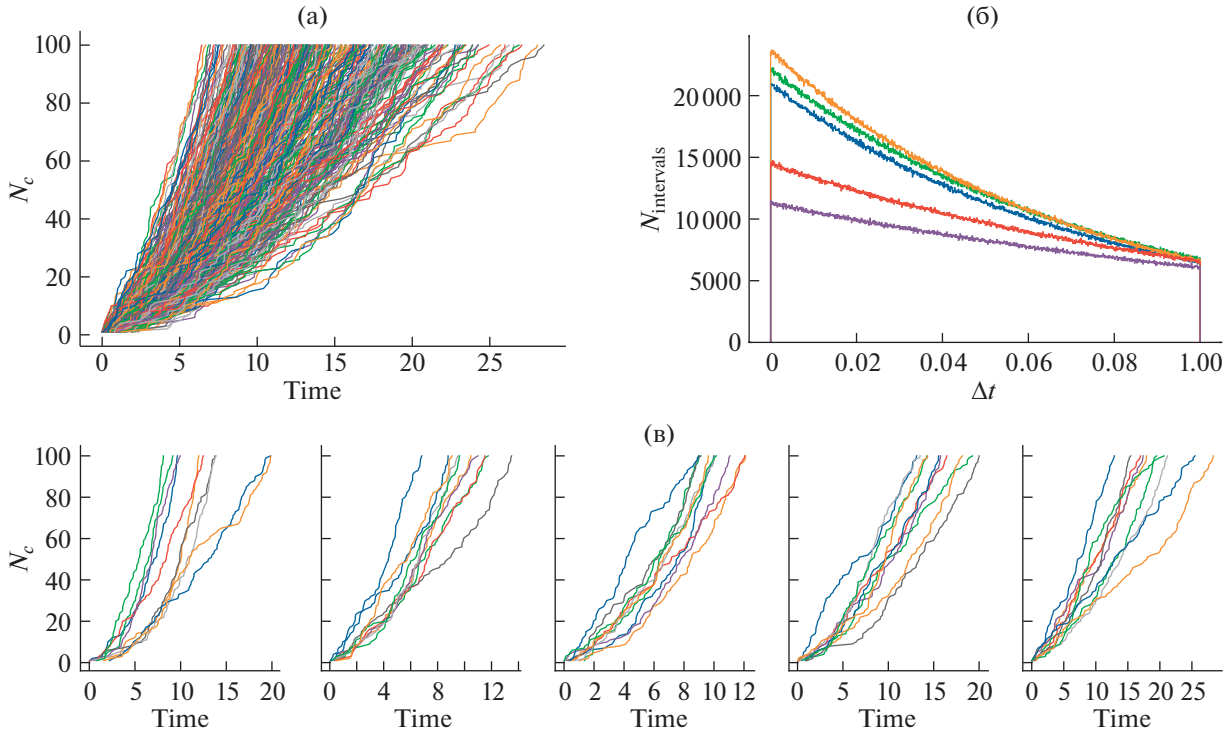

Рис. 3. Визуализация синтетических наборов данных. (а) функции счета для 50 последовательностей для всех кластеров, (б) гистограммы Δt между временем прибытия для каждого кластера, (в) функции счета для 10 последовательностей из каждого кластера.

всех экспериментах равен 2000. Результаты можно увидеть на табл. 2. Как видно, наш метод успешно обобщается на большие массивы данных.

Итоговая чистота по 15 запускам равна $0.576 \pm \pm 0.047$. Таким образом, как и ожидалось, выделить три кластера оказалось трудно. Однако метод успешно определил два других кластера. На рис. 4 показаны кривые обучения.

Сравнение модели

Огромный набор данных трудно использовать для сравнения моделей, поскольку другие модели не могут обрабатывать миллионы последовательностей за разумное время. Поэтому мы провели эксперименты на двенадцати меньших наборах данных. Мы сравнили нашу модель со следующими базовыми моделями:

• DMHP [46]. Модель смеси Дирихле для процессов Хоукса, объединяющая процессы Хоукса с EM-алгоритмом. У этой модели есть два недостатка – наличие гипотезы о самовозбуждении процессов и низкая скорость работы.

• THP [57]. Модель “Трансформер Хоукса” использует возможности трансформеров для прогнозирования различных поведенческих сценариев и достижения более высокой производительности. Однако эта модель не предполагает наличие кластеров и предполагает, что все последовательности генерируются на основе процесса с одинаковой интенсивностью. Чтобы применить эту модель для кластеризации, мы добавляем методом K-means поверх кодировок.

• Сверточный автокодировщик. Нейронная сеть автокодировщик, обученная восстанавливать последовательности, учитывая время как особенность. Чтобы применить эту модель для кластеризации, мы добавляем K-средние поверх кодировок. Не учитывает природу последовательности событий.

• Методы кластеризации из библиотеки tslearn. Не учитывают природу последовательности событий.

• K-means поверх признаков из библиотеки tsfresh. Не учитывает природу последовательности событий.

Таблица 2. Чистота по размеру набора данных при одинаковом максимальном объеме вычислений, равном 2000

Размер набора данных	Чистота
10^3	0.60
10^4	0.65
10^5	0.72

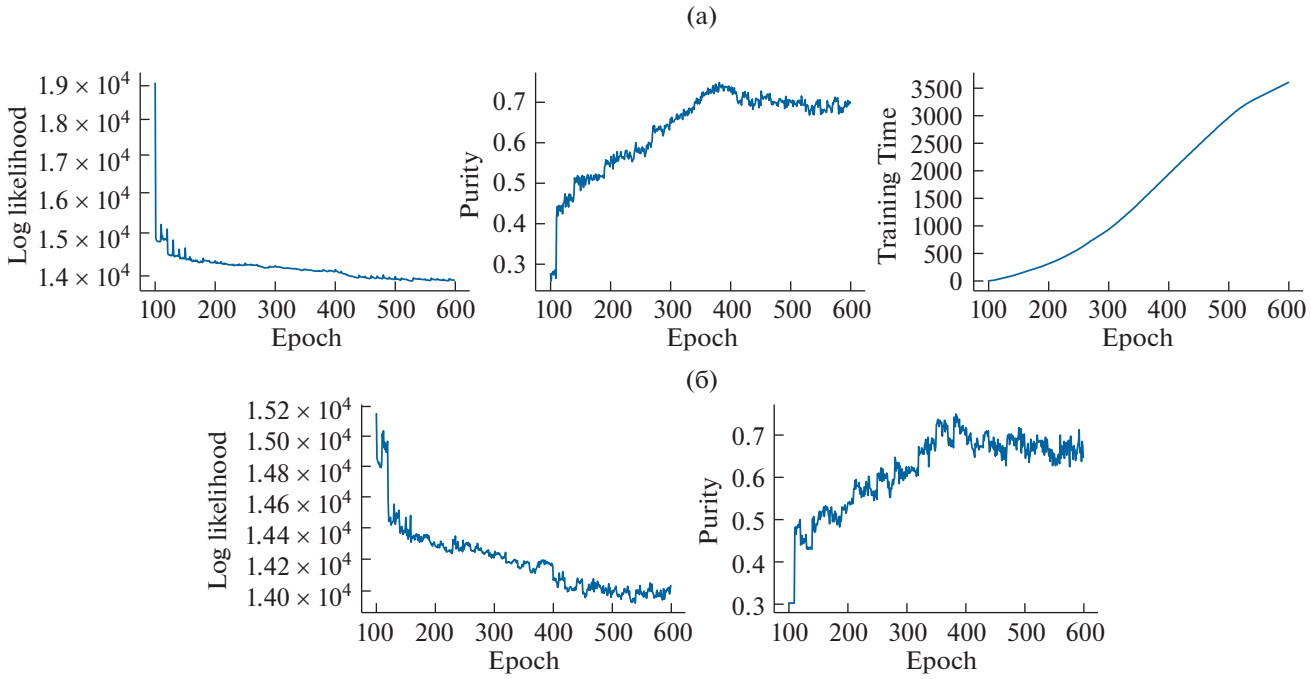

Рис. 4. Тренировочные кривые синтетического набора данных. (а) Слева направо: кривая потерь, чистоты и времени обучения в секундах, (б) кривая потерь при валидации и чистой валидации набора.

Результаты представлены в табл. 4 и 5. Как можно заметить, наш метод превосходит по выбранным показателям все базовые модели. Более того, несмотря на то, что все наборы данных были self-exciting, и, следовательно, можно было бы ожидать, что DMHP должен будет дать более высокие результаты, однако этого не произошло. Следует отметить, что DMHP также работает очень медленно (табл. 3) поэтому наш метод является лучшим как по качеству, так и по времени работы алгоритма.

Абляционное исследование

Модель зависит от различных метапараметров. Мы провели дополнительные эксперименты, чтобы показать, как эти параметры влияют на конечную производительность. Результаты можно найти на рис. 5. Мы можем сделать следующие выводы:

• Скорость обучения играет большую роль. Модель очень чувствительна. Если скорость обучения низкая, то модель склонна переобучаться на случайные метки. Чтобы предотвратить это, можно либо уменьшить скорость обучения еще больше и замедлить обучение, либо увеличить скорость обучения, чтобы достичь минимальной возможной потери, а затем уменьшать скорость обучения с помощью планировщиков, когда EM-алгоритм сходится к разумным кластерам. Мы выбираем второй вариант. Начальная скорость обучения составляет 0.1, а затем мы уменьшаем скорость обучения на плато с помощью множителя 0.5 и терпимости в 25 эпох.

• Начальное количество кластеров почти не влияет на конечную производительность.

• Количество шагов имеет некий минимальный порог, необходимый для достижения хороших результатов. Этот порог сопоставим с длиной

Таблица 3. Среднее время работы в секундах

Датасет	LAL	DMHP	Датасет	LAL	DMHP
K2_C5	2048	3044	sin_K4_C5	2706	236643
K3_C5	2305	49313	sin_K5_C5	2487	234628
K4_C5	2484	122645	trunc_K2_C5	1847	1540
K5_C5	2342	219504	trunc_K3_C5	1942	40850
sin_K2_C5	2078	71492	trunc_K4_C5	2201	77167
sin_K3_C5	2457	160122	trunc_K5_C5	2317	141553

Таблица 4. Результаты по метрикам чистоты синтетических датасетов для нашего метода (LAL) и базовых методов

Датасет	LAL (ours)	DMHP	Autoenc.	THP
K2_C5	0.97B ± 0.04	**1.00B ± 0.00**	0.65B ± 0.17	**1.00B ± 0.00**
K3_C5	0.85B ± 0.11	**0.92B ± 0.04**	0.64B ± 0.07	0.6B ± 0.01
K4_C5	**0.90B ± 0.07**	0.89B ± 0.14	0.45B ± 0.04	0.67B ± 0.06
K5_C5	**0.84B ± 0.09**	0.66B ± 0.06	0.46B ± 0.02	0.57B ± 0.04
sin_K2_C5	**0.99B ± 0.00**	0.93B ± 0.15	0.89B ± 0.03	0.89B ± 0.00
sin_K3_C5	**0.99B ± 0.01**	0.95B ± 0.02	0.80B ± 0.01	0.82B ± 0.00
sin_K4_C5	**0.92B ± 0.06**	0.81B ± 0.08	0.62B ± 0.07	0.55B ± 0.00
sin_K5_C5	**0.92B ± 0.05**	0.70B ± 0.03	0.47B ± 0.01	0.51B ± 0.01
trunc_K2_C5	**1.00B ± 0.00**	**1.00B ± 0.00**	**1.00B ± 0.00**	0.88B ± 0.17
trunc_K3_C5	**0.96B ± 0.01**	**0.96B ± 0.01**	0.59B ± 0.09	0.61B ± 0.00
trunc_K4_C5	**0.99B ± 0.00**	0.97B ± 0.05	0.75B ± 0.07	0.67B ± 0.02
trunc_K5_C5	**0.94B ± 0.06**	0.91B ± 0.08	0.64B ± 0.10	0.60B ± 0.02
Nr. of wins	**10**	4	1	1

Таблица 5. Результаты чистоты синтетических наборов данных для нашего метода (LAL) и базовых программ

Датасет	Tslearn Kshape	Tslearn Kmeans	Tsfresh Kmeans
K2_C5	0.76B ± 0.03	0.99B ± 0.00	0.94B ± 0.06
K3_C5	0.53B ± 0.05	0.80B ± 0.01	0.78B ± 0.08
K4_C5	0.36B ± 0.04	0.81B ± 0.02	0.53B ± 0.04
K5_C5	0.42B ± 0.03	0.55B ± 0.01	0.63B ± 0.04
sin_K2_C5	0.77B ± 0.10	0.93B ± 0.00	0.89B ± 0.04
sin_K3_C5	0.44B ± 0.08	0.84B ± 0.09	0.75B ± 0.13
sin_K4_C5	0.50B ± 0.05	0.59B ± 0.02	0.67B ± 0.04
sin_K5_C5	0.49B ± 0.04	0.58B ± 0.03	0.58B ± 0.05
trunc_K2_C5	0.75B ± 0.10	**1.00B ± 0.00**	0.78B ± 0.14
trunc_K3_C5	0.44B ± 0.06	0.80B ± 0.01	0.34B ± 0.00
trunc_K4_C5	0.41B ± 0.08	0.85B ± 0.10	0.35B ± 0.12
trunc_K5_C5	0.40B ± 0.03	0.76B ± 0.07	0.47B ± 0.14
Nr. of wins	0	1	0

последовательности. Дальнейшее увеличение разрешения не улучшает результаты.

• Верхнее ограничение кластеров также является значимым. Лучший верхний предел следует определить с использованием валидации.

Архитектуры RNN

Вместо LSTM-модели можно также рассмотреть стандартные RNN или GRU-архитектуры. В этом разделе мы приводим результаты сравнения этих моделей. С результатами можно ознакомиться в разделе табл. 6. Как можно заметить, наилучшие результаты показала архитектура на основе LSTM. Поэтому мы используем ее во всех экспериментах.

Ансамбли кластеров

Один из недостатков EM-алгоритма с нейронными сетями заключается в том, что он часто сходится к вырожденным решениям. В данной статье мы предложили методы улучшения стабильности, такие как случайное изменение числа кластеров и установку начальной скорости обучения на высокое значение. Однако также можно рассмотреть использование ансамблей кластеров. Мы использовали кластеризацию на основе матрицы сходства с использованием реализации из открытого репозитория на GitHub[1]. Мы использовали 20 обученных базовых моделей и применили спектральную кластеризацию поверх матрицы сходства. Результаты можно найти в табл. 7. Как можно заметить, применение ансамблей позволяет получить такое же или лучшее качество. Более того, для некоторых наборов данных улуч-

[1] https://github.com/jaumpedro214/posts/tree/main/ensamble_clustering

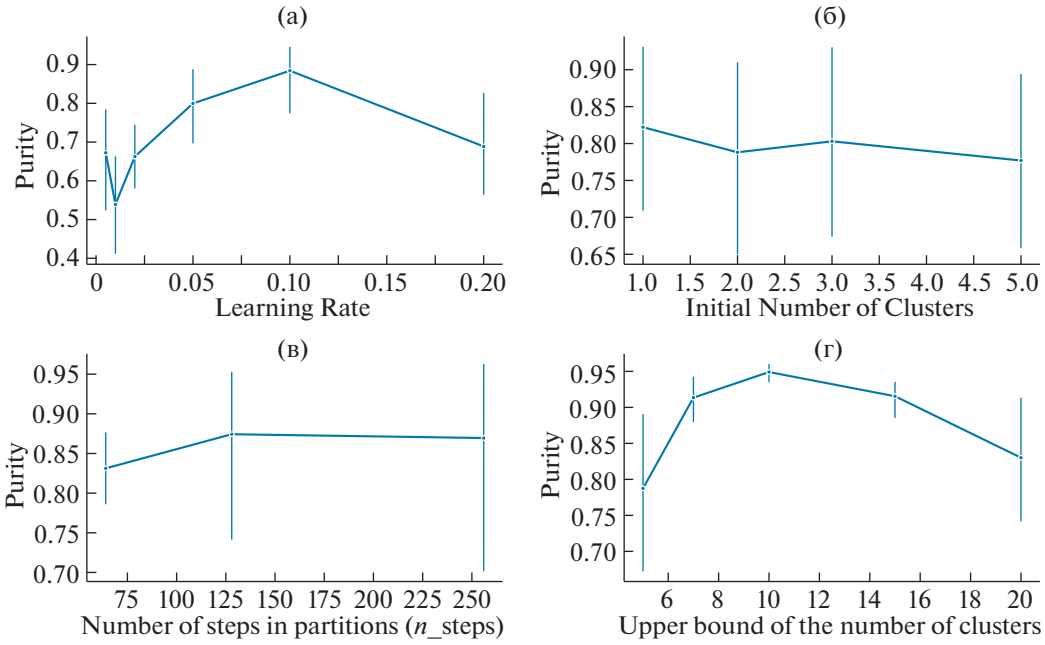

Рис. 5. Зависимость чистоты с доверительным интервалом в одно стандартное отклонение на наборе данных sin_K5_C5 при различных значениях скорости обучения (а), начального числа кластеров (б), n_steps (в), верхней границы числа кластеров (г).

шение значительное. Таким образом, можно использовать ансамбли для улучшения результатов. Однако обучение нейронных сетей может занимать много времени. Таким образом, существует компромисс между качеством и временем обучения.

5.2. Реальные наборы данных

Можно считать доказанной достаточно высокую производительность модели LAL на синтетических данных. Однако модель полезна, если справляется с задачами реального мира. Мы провели эксперименты на четырех наборах данных, связанных с реальными задачами.

Первый набор данных – это данные **IPTV** [6]. Он содержит данные о просмотрах 7100 пользователей IPTV от Shanghai Telecomm Inc. Авторы классифицировали телевизионные программы на 16 типов, и набор данных содержит данные о просмотрах с длиной сеанса более 20 мин. Мы кластеризуем пользователей на основе их данных о просмотре.

Второй набор данных – это данные **Age** [9]. Он содержит последовательности транзакций клиентов финансовых учреждений. Для каждого клиента у нас есть последовательность транзакций. Мы описываем каждую транзакцию дискретным кодом, который идентифицирует тип транзакции и код категории продавца, такие как книжный магазин, банкомат, аптека и другие. На основе истории транзакций мы категоризуем клиентов по возрастным группам. В данных предоставляется возрастной интервал каждого клиента с четырьмя различными интервалами. Мы сравниваем нашу модель кластеризации с настоящими возрастными интервалами.

Таблица 6. Чистота, полученная на синтетических наборах данных с помощью LAL, основанных на различных архитектурах (LSTM/RNN/GRU)

Датасет	LSTM	RNN	GRU
K2_C5	1.00 ± 0.00	1.00 ± 0.00	0.95 ± 0.02
K3_C5	0.99 ± 0.00	0.80 ± 0.00	0.82 ± 0.01
K4_C5	0.98 ± 0.06	0.91 ± 0.08	0.9 ± 0.05
K5_C5	0.94 ± 0.10	0.92 ± 0.09	0.92 ± 0.07
sin_K2_C5	0.99 ± 0.01	0.96 ± 0.03	0.99 ± 0.0
sin_K3_C5	0.99 ± 0.01	0.94 ± 0.01	0.91 ± 0.02
sin_K4_C5	0.93 ± 0.04	0.89 ± 0.05	0.88 ± 0.04
sin_K5_C5	0.92 ± 0.05	0.876 ± 0.08	0.88 ± 0.06
trunc_K2_C5	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
trunc_K3_C5	0.96 ± 0.01	0.94 ± 0.05	0.94 ± 0.02
trunc_K4_C5	0.99 ± 0.00	0.99 ± 0.00	0.99 ± 0.01
trunc_K5_C5	0.95 ± 0.04	0.97 ± 0.01	0.96 ± 0.01

Таблица 7. Сравнение чистоты одной модели LAL и ансамбля моделей LAL

Dataset	LAL	Ens.	Dataset	LAL	Ens.
K2_C5	0.97	1.00	sin_K4_C5	0.95	0.95
K3_C5	0.85	1.00	sin_K5_C5	0.92	0.96
K4_C5	0.90	1.00	trunc_K2_C5	1.00	1.00
K5_C5	0.84	1.00	trunc_K3_C5	0.96	0.97
sin_K2_C5	0.99	1.00	trunc_K4_C5	0.99	0.99
sin_K3_C5	0.99	0.99	trunc_K5_C5	0.94	0.96

Таблица 8. Результаты чистоты реальных наборов данных для нашего метода (LAL) и базовых методов

Model	Age	IPTV	Linkedin	Nr. of wins
LAL (ours)	0.34B ± 0.03	0.37B ± 0.01	0.26B ± 0.06	0
DMHP [4]6	**0.38B ± 0.02**	0.34B ± 0.03	0.31B ± 0.05	**1**
Autoenc.	0.31B ± 0.00	0.36B ± 0.00	0.21B ± 0.00	0
THP [57]	0.33B ± 0.00	**0.49B ± 0.00**	0.22B ± 0.01	**1**
Tslearn Kshape	0.26B ± 0.00	0.34B ± 0.01	0.20B ± 0.00	0
Tslearn Kmeans	0.26B ± 0.00	0.35B ± 0.02	0.20B ± 0.00	0
Tsfresh Kmeans	0.26B ± 0.00	0.38B ± 0.01	**0.44B ± 0.02**	**1**

Третий набор данных – это данные **LinkedIn** [46]. Он содержит историю работодателей пользователей. Всего в наборе представлены 82 различные компании с различными типами событий и общим числом 2439 пользователей. На основе этих данных мы можем категоризировать пользователей по различным профессиональным областям. Для каждого пользователя доступна информация о секторах индустрии для сравнения их с реальными. Последовательности событий в этих трех наборах данных имеют сильные запускающие паттерны, которые мы можем смоделировать с помощью различных процессов Хоукса.

Четвертый набор данных – это данные о продуктах **Amazon**[2]. Он состоит из около 100 тыс. последовательностей пользовательских отзывов о продуктах и категорий продуктов. Все продукты разделены на 8 различных категорий. Размер этого набора данных позволяет нам также продемонстрировать масштабируемость модели на данных реального мира.

Исследования, основанные на кластеризации. Основная цель данной статьи – получить хорошие кластеры. Поэтому первый фокус состоит в проверке кластеров на наборе данных реального мира с помощью определенной метрики. Однако есть различные способы, как можно кластеризовать точки данных, например, людей можно разделить по возрасту, полу или области деятельности. Все эти разделения будут допустимыми, и выбор должен зависеть от конкретной проблемы. Однако эти исследования кластеризации будут полезны для интерпретации результатов и могут предоставить ценные идеи. Для проведения этих исследований необходимо присвоить псевдометки точкам данных для вычисления метрик. Мы провели эксперименты на трех наборах данных реального мира: IPTV, Age и Linkedin, с использованием следующих меток:

- IPTV – тип последнего события
- Age – четыре возрастных интервала
- LinkedIn – секторы индустрии

Набор данных Amazon использовался для вторичной задачи и для демонстрации масштабируемости метода. Использовалась мера качества – чистота (purity).

Результаты представлены в табл. 8. Как видно, наша модель не является лучшей с точки зрения чистоты. Тем не менее рассмотрим кривые обучения более подробно на рис. 6.

После тренировки видно, что чистота не самая лучшая. Во время первой половины тренировки чистота становится выше, а потом становится ниже. Это означает, что существует корреляция между изученным кластером и псевдо-метками, и в первой половине обучения мы ловим некоторые зависимости, которые также можно найти в псевдо-метках. Однако дальнейшее обучение приводит к кластерам, которые отличаются от псевдометок и позволяют повысить степень соответствия закону интенсивности. Мы построили график 2 компонентов t-SNE для изученных

[2] http://jmcauley.ucsd.edu/data/amazon/links.html

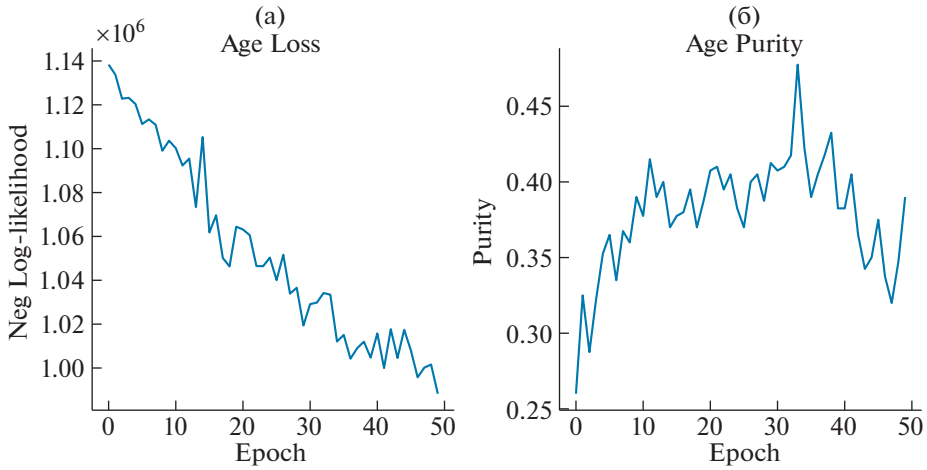

Рис. 6. Кривая потерь и чистоты по эпохам на наборе данных Age для метода LAL.

представлений, чтобы показать, что можно получить различимые кластеры на реальных данных. В качестве представлений мы будем использовать состояние ячейки конечного слоя. Обоснование такого выбора сводится к тому, что состояние ячейки отвечает за память, и, таким образом, ожидается, что она лучше представляет данные в целом. Графики для набора данных Age можно увидеть на рис. 7.

Как можно заметить, существует четыре различных кластера. Однако они не очень хорошо отражают возрастные категории. Вот почему желательна дополнительная проверка, основанная на прикладной задаче.

Прикладная задача. Кластеры сами по себе не очень полезны до того момента, пока мы не можем использовать их для прогнозирования чего-то более конкретного. Кроме того, мы можем рассматривать и другие подходы к кластеризации с различными разбиениями. Таким образом, наша цель – представить задачу, аналогичную выбору аудитории с использованием сгенерированных кластеров, и тем самым показать, как можно использовать LAL в маркетинге.

Далее, чтобы показать устойчивость LAL, мы рассматриваем обучение представлений без учителя 35 сценариев в рамках изучения представлений, поскольку наша кластеризация имеет промежуточные представления.

Поэтому мы используем следующий метод проверки качества полученных представлений. Мы рассматриваем практические задачи, имитирующие выбор аудитории для подтверждения полученных результатов. В частности, мы рассматриваем возможность предсказания наиболее часто встречающегося типа события во временном окне, равном 10% длины последовательности. Это эквивалентно предоставлению интересов для аудитории или категории интересов. Например, маркетолог может нацелиться на всех, кто интересуется книгами или детскими игрушками.

Учитывая это, мы рассматриваем последнее состояние ячейки LSTM как представление последовательности, как это было сделано при исследовании t-SNE-представления. Состояние ячейки – это память LSTM и может обладать достаточной информацией о всей последовательности. Многие работы следуют этим подходам для последовательной обработки данных в целом и кластеризации в частности.

Мы сравниваем LAL с другими современными подходами. Наша цель – показать, как работает система в сравнении с теми архитектурами, которые многие специалисты могут считать более современными, или подходами, вызвавшими в последние годы больший интерес сообщества, чем LSTM.

В частности, мы используем трансформер Хоукса [57], или THP, и мультимодальный подход к вычислению представлений, использующий комбинацию последовательностей и текста, там, где это применимо, и TS-fresh библиотеку для получения особенностей последовательностей в качестве базовых представлений временных рядов для сравнения. Примечание: мультимодальность требует дополнительной модальности и поэтому может быть применена только для данных Amazon.

В случае THP мы используем выход кодирующей модели с механизмом self-attention, после чего пропускаем его через слой LSTM для получения представлений последовательности.

Для второго сравнения, multimodality, мы используем идеи мультимодальности, например [49] для изображений и текста. В нашем случае мы рассматриваем временные ряды и текст для создания представлений, которые мы объединя-

ем. Для создания представлений временных рядов мы используем рекуррентные временные точечные процессы с отмеченной точкой [31]. Далее, мы создаем представления текста с помощью архитектуры [19]. После того как мы получили оба представления, мы объединяем их.

Последний подход (получение особенностей с помощью TS fresh) применяется для того, чтобы показать, что простых признаков, которые можно использовать в качестве представления последовательности, недостаточно, чтобы превзойти наш метод, и что представления, которые дает метод LAL, – лучше.

В среде сообщества специалистов по прикладным задачам машинного обучения распространен прием, при котором второй шаг в выборе аудитории выполняется с помощью логистической регрессии. Мы следуем этой практике и обучаем логистическую регрессию поверх всех встраиваний. Мы рассчитываем результаты и сравниваем оценки ROC AUC для задач “one-vs-all” для каждого типа событий. Наша методология схожа с [12].

Результаты выполнения прикладной задачи можно увидеть в табл. 9. Как можно заметить, наш метод показал превосходную производительность при решении прикладной задачи, что позволяет использовать его в бизнес-приложениях.

На *данных о товарах Amazon* мы используем около 100 тыс. точек для обучения. Аналогичным образом мы рассматриваем подмножество из 7.5 тыс. последовательностей для проверки результатов и сравнения LAL с THP и Multimodality. Таким образом, мы обучаем классификаторы (логистические регрессии) на основе встраиваний LAL, THP, Multimodality и TSFresh. Мы построили ROC-кривые для каждого типа события в рис. 8.

Таблица 9. Показатели ROC AUC на реальных наборах данных. Время обучения THP и Multimodality оказалось слишком большим для завершения обучения на данных Amazon 100k. Метод мультимодальности применим только к данным Amazon

Датасет	LAL (ours)	THP	Multimodality	TSFresh
Age	**0.983**	0.935	–	0.927
IPTV	**1.000**	0.969	–	0.924
LinkedIn	**0.887**	0.741	–	0.5
Amazon 7.5k	**0.959**	0.803	0.538	0.459
Amazon 100k	**0.990**	–	–	0.542
Nr. of wins	**5**	0	0	0

Мы наблюдаем, что мультимодальный подход показывает неоптимальные результаты. Дальнейший анализ показывает, что мультимодальность может быть менее применимой для текстов и временных, чем для комбинации изображений и текстов. Таким образом, модель требует доработки.

Помимо этого, мы оценили производительность LAL на полном *наборе данных о товарах Amazon* и привели ее табл. 9. Как видно, наш метод успешно обобщается на достаточно обширную базу данных. В приложении мы приводим ROC-кривые для этого набора данных.

Для других наборов данных Multimodality неприменима, однако для них можно использовать THP и TS fresh, а также метод LAL. Мы вычисляем ROC-AUC метрику для всех методов табл. 9. Становится очевидным, что LAL превосходит другие методы и обеспечивает лучшее встраивание.

Стоит отметить, что результаты достаточно хороши для прогнозирования “one vs all”; однако

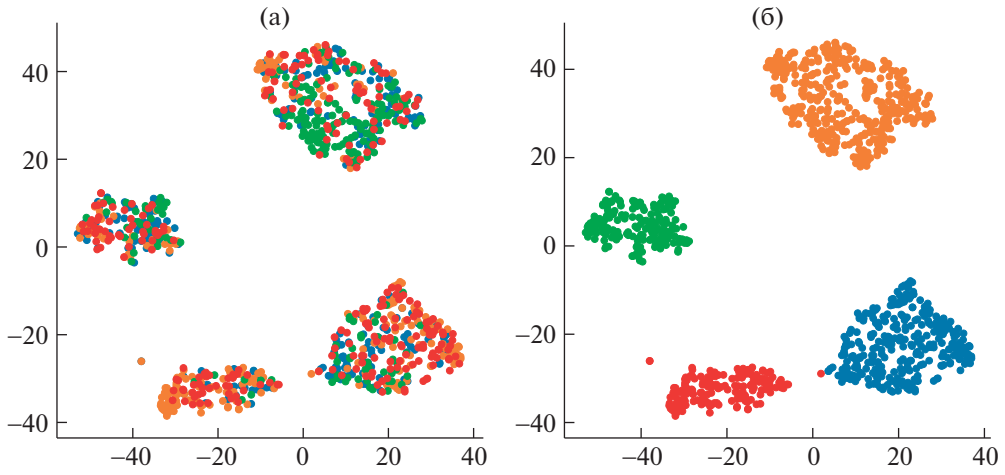

Рис. 7. t-SNE для представлений на основе LAL для кластеров из реального мира, кластеров из реального набора данных Age, закрашенных реальными метками (а) и выученными метками (б).

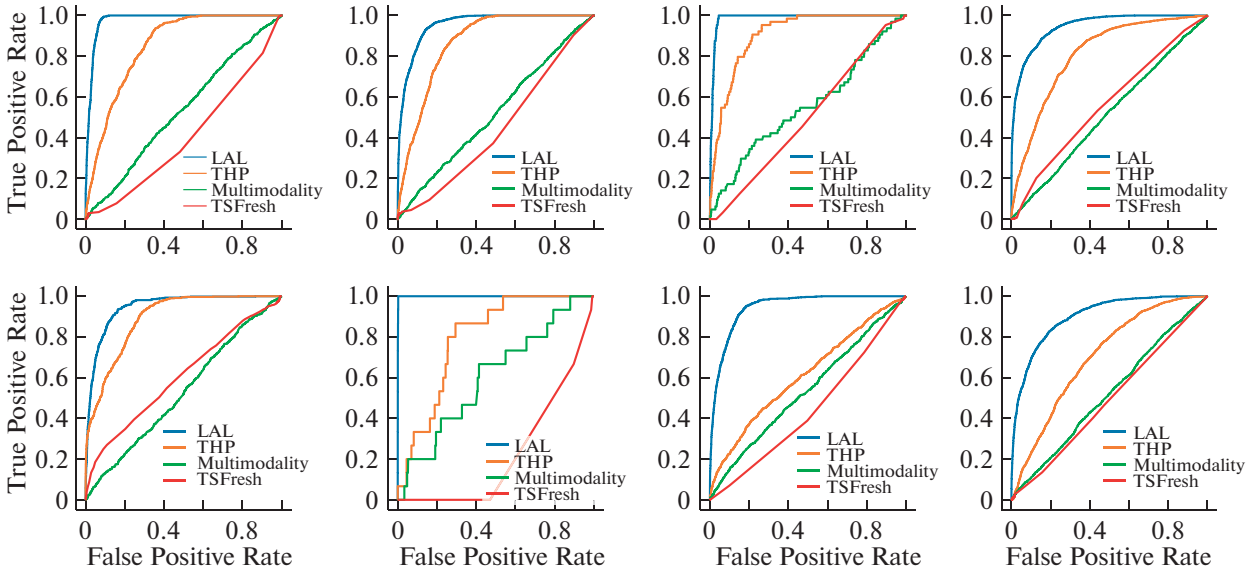

Рис. 8. ROC-кривые для всех типов событий при задаче one-versus-all для наборов данных Amazon с 7.5 тыс. точек.

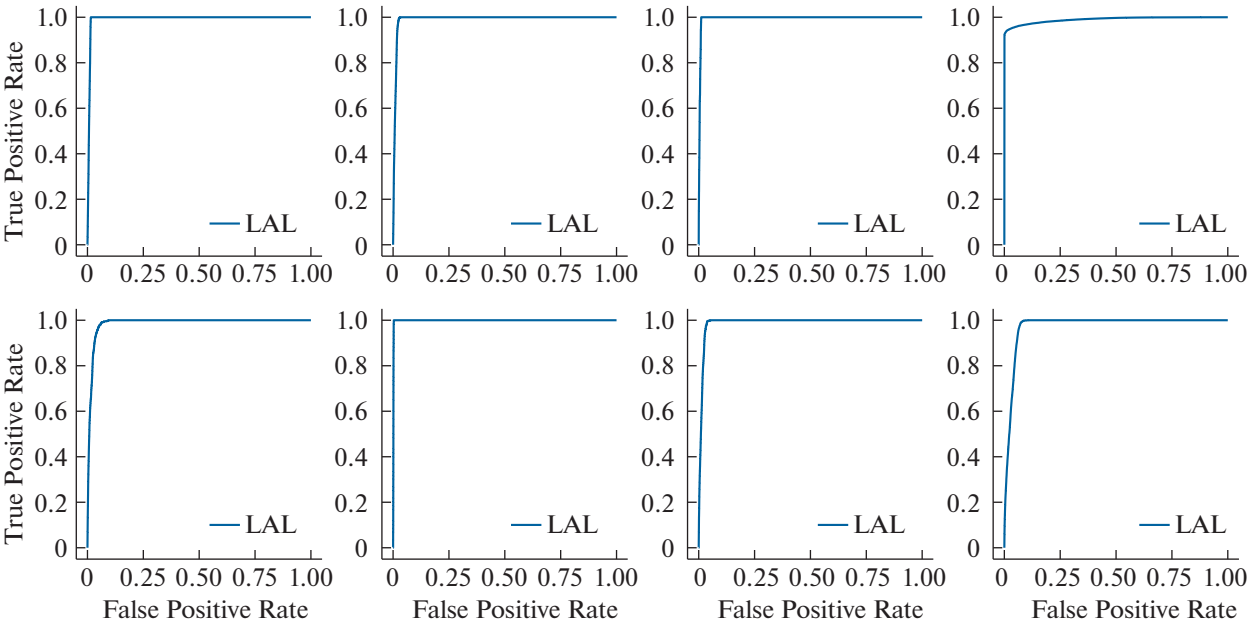

Рис. 9. ROC-кривые для всех типов событий в датасете 100k Amazon.

модели сложно предсказать все классы одновременно. Нам следует рассмотреть более сложные модели для второго этапа отбора аудитории в качестве потенциальной альтернативы для повышения результатов нашей работы.

6. ОБСУЖДЕНИЕ

Подводя итог, можно отметить, что LAL может работать не только с проприетарными данными компании. Мы показываем его производительность на синтетическом наборе данных и четырех реальных наборах данных. Также мы предоставляем последующую задачу, сравнимую со вторым этапом отбора аудитории, который обычно встречается в сообществе практиков. Для решения этой задачи мы также получаем надежные результаты. Однако для некоторых классов кривые ROC близки. Более того, мы проводим сравнительный анализ с THP, более сложной моделью, чем LSTM, основанной на архитектуре трансформеров. Интуитивно мы могли бы ожидать, что мо-

дель THP будет работать лучше. Однако это было не так. Одной из причин такого поведения является то, что THP является бескластерной моделью. Он изучает представления, но пропускает внутренние кластерные зависимости. В нашем LAL используется более простая модель, но она включает кластеры. Таким образом, достигается компромисс между учетом кластеров и использованием более сложной модели. Для представленной задачи кластеры оказываются решающими. Сложность модели здесь не имеет значения.

Стоит отметить, что простое извлечение признаков с использованием библиотек пространственных временных рядов, таких как TS Fresh, также не обеспечивает такого качества представления, как наш метод.

Сторонний наблюдатель мог бы предположить, что включение THP вместо LSTM или другой архитектуры, разработанной на основе трансформеров, в LAL должно повысить его производительность. Однако внутренние эксперименты показывают, что замена архитектуры значительно снижает производительность.

Мы объясняем это наблюдение тем фактом, что нейронная сеть, использующая модель трансформеров, значительно лучше улавливает нелинейности. При этом алгоритм EM может завершаться на плохих локальных минимумах.

Модель переобучается на случайных метках в течение первых эпох и не может выйти за пределы этого минимума.

Простого решения этой проблемы не существует. Мы пытаемся использовать регуляризацию для успешного использования более сложных моделей.

Чтобы улучшить качество кластеризации, можно также использовать кластерные ансамбли. Это позволяет добиться еще более высокой производительности. Однако это требует дополнительных вычислений.

7. ЗАКЛЮЧЕНИЕ

В этой работе предлагается система, предназначенная для кластеризации последовательностей событий с целью отбора аудитории для маркетологов. Наш метод сочетает в себе модель LSTM, обученную с помощью нашего варианта алгоритма EM. Кроме того, мы разработали процедуру двойного батчинга, которая позволяет обучать модель на наборах данных произвольных размеров. Мы подчеркиваем способность нашего подхода давать конкурентоспособные результаты: для общедоступных наборов данных наблюдается более высокое качество по сравнению со стандартными подходами. Далее мы показываем, что мы можем использовать наши модельные представления для решения практических задач, эквивалентных второму этапу отбора аудитории. В представленной задаче наш метод превосходит более сложные современные архитектуры, а также простое извлечение признаков для временных рядов.

Приложение A

ДОПОЛНИТЕЛЬНЫЕ РИСУНКИ

Кривая ROC 100k

На рис. 10 можно найти кривые ROC для всех типов событий для метода LAL. Мы видим явное улучшение по сравнению со случаем, когда используется всего 7.5 тыс. объектов.

БЛАГОДАРНОСТИ

Мы благодарим команду Choco и группу ADASE в Сколтехе за их поддержку. Работа Всеволода Грабаря, Алексея Зайцева и Евгения Бурнаева была поддержана Аналитическим центром Сколтеха (соглашение о субсидировании 000000D730321P5Q0002, грант № 70-2021-00145 от 02.11.2021).

ИСТОЧНИК ФИНАНСИРОВАНИЯ

Работа Всеволода Грабаря, Алексея Зайцева и Евгения Бурнаева была поддержана Аналитическим центром Сколтеха (соглашение о субсидировании 000000D730321P5Q0002, грант № 70-2021-00145 от 02.11.2021).

СПИСОК ЛИТЕРАТУРЫ

1. *Mathilde Caron et al.* “Unsupervised learning of visual features by contrasting cluster assignments”. B: Advances in Neural Information Processing Systems. 2020. V. 33. P. 9912–9924.
2. *Gromit Yeuk-Yin Chan et al.* “Interactive Audience Expansion On Large Scale Online Visitor Data”. B: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. New York, NY, USA: Association for Computing Machinery, авг. 2021. P. 2621–2631.
3. *Daley D., Vere-Jones D.* “An introduction to the theory of point processes. Vol. I: Elementary theory and methods. 2nd ed”. B: Vol. 1 (янв. 2003). https://doi.org/10.1007/b97277
4. *Arthur P. Dempster, NanMLaird, Donald B. Rubin.* “Maximum likelihood from incomplete data via the EM algorithm”. B: Journal of the Royal Statistical Society: Series B (Methodological). 1977. V. 39.1. P. 1–22.
5. *Stephanie deWet, Jiafan Ou.* “Finding Users Who Act Alike: Transfer Learning for Expanding Advertiser Audiences”. B: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. KDD ’19. Anchorage, AK, USA: Associ-

ation for Computing Machinery, июль 2019. P. 2251–2259.

6. *Dixin Luo et al.* "You Are What You Watch and When You Watch: Inferring Household Structures From IPTV Viewing Data". B: IEEE Transactions on Broadcasting, vol. 60, no. 1, pp. 61–72, March 2014. 2014.
7. *Khoa D. Doan, Pranjul Yadav, Chandan K Reddy*. "Adversarial Factorization Autoencoder for Look-alike Modeling". B: Proceedings of the 28th ACM International Conference on Information and Knowledge Management. CIKM '19. Beijing, China: Association for Computing Machinery, нояб. 2019. P. 2803–2812.
8. *Faraone M.F. et al.* "Using context to improve the effectiveness of segmentation and targeting in e-commerce". B: Expert Systems with Applications. 2012. V. 39.9. P. 8439–8451. issn: 0957-4174. https://doi.org/10.1016/j.eswa.2012.01.174. url: https://www.sciencedirect.com/science/article/pii/S0957417412002023.
9. *Fursov I. et al.* "Gradient-based adversarial attacks on categorical sequence models via traversing an embedded world". B: AIST. 2020.
10. *Fursov Ivan et al.* "Sequence Embeddings Help Detect Insurance Fraud". B: IEEE Access. 2022. V. 10. P. 32060–32074.
11. *Felix A. Gers, Jürgen Schmidhuber,* Fred Cummins. "Learning to forget: Continual prediction with LSTM". B. 1999.
12. *Jean-Bastien Grill et al.* "Bootstrap your own latent-a new approach to self-supervised learning". B: Advances in Neural Information Processing Systems. 2020. V. 33. P. 21271–21284.
13. *Luo Haiyan et al.* Methods and systems for near real-time lookalike audience expansion in ads targeting. en. https://patents.justia.com/patent/10853847. Accessed: 2022-2-7.
14. *Alan G. Hawkes.* "Spectra of some self-exciting and mutually exciting point processes". B: Biometrika. 1971. V. 58.1. P. 83–90.
15. *Kaiming He et al.* "Masked autoencoders are scalable vision learners". B: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022. P. 16000–16009.
16. *Sergey Ioffe, Christian Szegedy.* "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". B: (февр. 2015).
17. *Ashish Jaiswal et al.* "A survey on contrastive self-supervised learning". B: Technologies. 2020. V. 9.1. P. 2.
18. *Jinling Jiang et al.* Comprehensive audience expansion based on end-toend neural prediction. http://ceur-ws.org/Vol-2410/paper12.pdf. Accessed: 2022-2-8.
19. *Yoon Kim.* "Convolutional Neural Networks for Sentence Classification". B: EMNLP. 2014.
20. *Dennis Koehn, Stefan Lessmann, Markus Schaal.* "Predicting online shopping behaviour from clickstream data using deep learning". B: Expert Systems with Applications. 2020. V. 150. P. 113342. issn: 0957-4174. https://doi.org/10.1016/j.eswa.2020.113342. url: https://www.sciencedirect.com/science/article/pii/S0957417420301676.
21. *Haishan Liu et al.* "Audience Expansion for Online Social Network Advertising". B: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2016.
22. *Haishan Liu et al.* "Audience Expansion for Online Social Network Advertising". B: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '16. San Francisco, California, USA: Association for Computing Machinery, авг. 2016. P. 165–174.
23. *Yudan Liu et al.* "Real-time Attention Based Look-alike Model for Recommender System". B: (июнь 2019). arXiv: 1906.05022 [cs.IR].
24. *Zhining Liu et al.* "Two-Stage Audience Expansion for Financial Targeting in Marketing". B: Proceedings of the 29th ACM International Conference on Information & Knowledge Management. CIKM '20. Virtual Event, Ireland: Association for Computing Machinery, окт. 2020. P. 2629–2636.
25. *Qiang Ma, Musen Wen, Datong Chen*. A sub-linear, massive-scale lookalike audience extension system. http://proceedings.mlr.press/v53/ma16.pdf. Accessed: 2022-2-7.
26. *Qiang Ma et al.* "Score Look-alike Audiences". B: ().
27. *Ashish Mangalampalli et al.* "A feature-pair-based associative classification approach to look-alike modeling for conversion-oriented user-targeting in tail campaigns". B: Proceedings of the 20th international conference companion on World wide web.WWW'11. Hyderabad, India: Association for Computing Machinery, март 2011. P. 85–86.
28. *Ashish Mangalampalli et al.* "A feature-pair-based associative classification approach to look-alike modeling for conversion-oriented user-targeting in tail campaigns". B: Proceedings of the 20th international conference companion on World wide web – WWW '11. Hyderabad, India: ACM Press, 2011.
29. *Hongyuan Mei, Jason Eisner.* "The Neural Hawkes Process: A Neurally Self-Modulating Multivariate Point Process". B: NeurIPS. 2017.
30. *Lin Miao, Mark Last, Marina Litvak*. "Tracking social media during the COVID-19 pandemic: The case study of lockdown in New York State". B: Expert Systems with Applications. 2022. V. 187. P. 115797. issn: 0957-4174. https://doi.org/10.1016/j.eswa.2021.115797. url: https://www.sciencedirect.com/science/article/pii/S0957417421011659.
31. *Du Nan et al.* "Recurrent Marked Temporal Point Processes: Embedding Event History to Vector". B: KDD'16. 2016.
32. *Aaron van den Oord et al.* "WaveNet: A Generative Model for Raw Audio". B: 9th ISCA Speech Synthesis Workshop.

33. *Sandeep Pandey et al.* “Learning to target: what works for behavioral targeting”. B: Proceedings of the 20th ACM international conference on Information and knowledge management. CIKM ’11. Glasgow, Scotland, UK: Association for Computing Machinery, окт. 2011. P. 1805–1814.

34. *Perlich C. et al.* “Machine learning for targeted display advertising: Transfer learning in action”. B: ().

35. *Yan Qu et al.* “Systems and methods for generating expanded user segments”. 8655695. Февр. 2014.

36. *Ernest Kirubakaran Selvaraj.* Multigraph-Lookalike. en.

37. *Ernest Kirubakaran Selvaraj et al.* “Multigraph Approach Towards a Scalable, Robust look-alike Audience Extension System”. B. 2021.

38. *Oleksandr Shchur et al.* “Neural temporal point processes: A review”. B: arXiv preprint arXiv:2104.03528. 2021.

39. *Jianqiang Shen, Sahin Cem Geyik, Ali Dasdan.* “Effective Audience Extension in Online Advertising”. B: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD ’15. Sydney, NSW, Australia: Association for Computing Machinery, авг. 2015. P. 2099–2108.

40. *Hui Shi et al.* “Continuous CNN for nonuniform time series”. B: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE. 2021. P. 3550–3554.

41. *Victoria Snorovikhina, Alexey Zaytsev.* “Unsupervised anomaly detection for discrete sequence healthcare data”. B: International Conference on Analysis of Images, Social Networks and Texts. Springer. 2020. P. 391–403.

42. *Yi Tay et al.* “Efficient transformers: A survey”. B: ACM Computing Surveys. 2022. V. 55.6. P. 1–28.

43. *Yi Tay et al.* “Long range arena: A benchmark for efficient transformers”. B: arXiv preprint arXiv:2011.04006. 2020.

44. *Nikolaos Tziortziotis et al.* “Audience expansion based on user browsing history”. B: 2021 International Joint Conference on Neural Networks (IJCNN). 2021. P. 1–8. https://doi.org/10.1109/IJCNN52387.2021.9533392.

45. *Yansong Wang et al.* “CasSeqGCN: Combining network structure and temporal sequence to predict information cascades”. B: Expert Systems with Applications. 2022. V. 206. P. 117693. issn: 0957-4174. https://doi.org/10.1016/j.eswa.2022.117693. url: https://www.sciencedirect.com/science/article/pii/S095741742200985X.

46. *Hongteng Xu, Hongyuan Zha.* “A Dirichlet mixture model of Hawkes processes for event sequence clustering”. B: Advances in Neural Information Processing Systems. 2017. P. 1354–1363.

47. *Junchi Yan.* “Recent advance in temporal point process: from machine learning perspective”. B: SJTU Technical Report. 2019.

48. *Carl Yang et al.* “I Know You’ll Be Back: Interpretable New User Clustering and Churn Prediction on a Mobile Social Application”. B: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. KDD ’18. London, United Kingdom: Association for Computing Machinery, июль 2018. P. 914–922.

49. *Wang Yaqing et al.* “EANN: Event Adversarial Neural Networks for Multi-Modal Fake News Detection”. B: KDD. 2018.

50. *James Zhang et al.* “Dynamic time warp-based clustering: Application of machine learning algorithms to simulation input modelling”. B: Expert Systems with Applications. 2021. V. 186. P. 115684. issn: 0957-4174. https://doi.org/10.1016/j.eswa.2021.115684. url: https://www.sciencedirect.com/science/article/pii/S0957417421010691.

51. *Yunhao Zhang et al.* “Learning mixture of neural temporal point processes for multi-dimensional event sequence clustering”. B: Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, Vienna, Austria. 2022. P. 23–29.

52. *Zhihua Zhang et al.* “Learning a Multivariate Gaussian Mixture Model with the Reversible Jump MCMC Algorithm”. B: Statistics and Computing 14 (апр. 2004). https://doi.org/10.1023/B:STCO.0000039484.36470.41

53. *Yongchun Zhu et al.* “Learning to Expand Audience via Meta Hybrid Experts and Critics for Recommendation and Advertising”. B: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. New York, NY, USA: Association for Computing Machinery, авг. 2021. P. 4005–4013.

54. *Chenyi Zhuang et al.* “Hubble: An Industrial System for Audience Expansion in Mobile Marketing”. B: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. KDD ’20. Virtual Event, CA, USA: Association for Computing Machinery, 2020. P. 2455–2463. isbn: 9781450379984. https://doi.org/10.1145/3394486.3403295. url: https://doi.org/10.1145/3394486.3403295

55. *Chenyi Zhuang et al.* “Hubble: An Industrial System for Audience Expansion in Mobile Marketing”. B: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. New York, NY, USA: Association for Computing Machinery, авг. 2020. P. 2455–2463.

56. *Jinfeng Zhuang et al.* “PinText 2: Attentive bag of annotations embedding”. B: ().

57. *Simiao Zuo et al.* “Transformer Hawkes process”. B: International conference on machine learning. PMLR. 2020. P. 11692–11702.

NO TWO USERS ARE ALIKE: GENERATING AUDIENCES WITH NEURAL CLUSTERING FOR TEMPORAL POINT PROCESSES

V. Zhuzhel[a], V. Grabar[a], N. Kaploukhaya[d], R. Rivera-Castro[b,c,d], L. Mironova [a], A. Zaytsev[a], and E. Burnaev[a]

[a]*Skolkovo Institute of Science and Technology, Moscow, Russia*
[b]*Choco Communications, Berlin, Germany*
[c]*Center for Digital Technology and Management, Munich, Germany*
[d]*Research done while was working at Skolkovo Institute of Science and Technology, Moscow, Russia*
Presented by Academician of the RAS A.I. Avetisyan

Identifying the right user to target is a common problem for different Internet platforms. Although numerous systems address this task, they are heavily tailored for specific environments and settings. It is challenging for practitioners to apply these findings to their problems. The reason is that most systems are designed for settings with millions of highly active users and with personal information, as is the case in social networks or other services with high virality. There exists a gap in the literature for systems that are for medium-sized data and where the only data available are the event sequences of a user. It motivates us to present Look-A-Liker (LAL) as an unsupervised deep cluster system. It uses temporal point processes to identify similar users for targeting tasks. We use data from the leading Internet marketplace for the gastronomic sector for experiments. LAL generalizes beyond proprietary data. Using event sequences of users, it is possible to obtain state-of-the-art results compared to novel methods such as Transformer architectures and multimodal learning. Our approach produces the up to 20% ROC AUC score improvement on real-world datasets from 0.803 to 0.959. Although LAL focuses on hundreds of thousands of sequences, we show how it quickly expands to millions of user sequences. We provide a fully reproducible implementation with code and datasets in https://github.com/adasegroup/sequence_clusterers.

Keywords: applications, clustering, unsupervised, temporal point processes