Выявление взаимозависимости между метками с использованием алгоритма, основанного на собственном внимании в задаче классификации с несколькими метками.

Боева Галина
Антиплагиат
Сколтех
boeva.gl@phystech.edu

Консультант: к.ф.-м.н. Грабовой Андрей
Антиплагиат
grabovoy.av@phystech.edu

Эксперт: к.ф.-м.н. Зайцев Алексей
Сколтех
a.zaytsev@skoltech.ru

28 марта 2024 г.

Аннотация

Большая часть доступной пользовательской информации может быть представлена в виде последовательности событий с временными метками. Каждому событию присваивается набор категориальных меток, будущая структура которых представляет большой интерес. Это задача прогнозирования временных наборов для последовательных данных. Современные подходы фокусируются на архитектуре преобразования последовательных данных, используя собственного внимания(“self-attention”) к элементам в последовательности. В этом случае мы учитываем временные взаимодействия событий, но теряем информацию о взаимозависимостях меток. Мотивированные этим недостатком, мы предлагаем использовать механизм собственного внимания(“self-attention”) к меткам, предшествующим прогнозируемому шагу. Поскольку наш подход представляет собой сеть внимания к меткам, мы называем ее LANET. Мы также обосновываем этот метод агрегирования, он положительно влияет на интенсивность события, предполагая, что мы используем стандартный вид интенсивности, предполагая работу с базовым процессом Хоукса.

Ключевые слова временные ряды · взаимосвязь меток

1 Введение

Классификация с несколькими метками является более естественной, чем бинарная или многоклассовая классификация, поскольку все, что окружает нас в реальном мире, обычно описывается несколькими метками [1]. Та же логика может быть перенесена на последовательность событий с отметками времени. События в последовательности, как правило, характеризуются несколькими категориальными значениями вместо одной. Существует множество подходов к классификации с несколькими метками в компьютерном зрении [2], обработке естественного языка [3] или классической структуре табличных данных [4]. Однако постановке задачи с несколькими метками для последовательностей событий, как

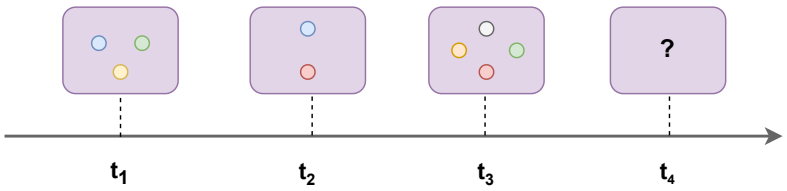

Рис. 1: На рисунке показано визуальное представление постановки задачи. Наша модель должна предсказать метки для момента времени t_4, учитывая историю предыдущих наборов меток. Требуется предсказать несколько меток, так что это определение задачи классификации с несколькими метками.

правило, уделяется меньше внимания. Итак, мы стремимся противостоять такому недостатку внимания и решить проблему предсказания набора меток для последовательных данных с временными метками. Важно отметить, что модель должна предсказывать набор меток, соответствующих следующему шагу, принимая во внимание содержимое предыдущих групп меток для последовательности событий, связанных с объектом(Рисунок 1).

Взаимодействие между состояниями объекта в разные временные метки имеет важное значение для решения задач с последовательными данными [5]. Следовательно, выразительные и мощные модели должны быть способны изучать такие взаимодействия. Несколько архитектур нейронных сетей, таких как трансформеры или рекуррентные нейронные сети, способны делать это. Например, трансформер напрямую определяет механизм внимания, который измеряет, как связаны различные временные метки в последовательности. Однако применение современных методов глубокого обучения ограничено [6], и они в первую очередь сосредоточены на прогнозировании меток для последовательности в целом.

Основные подходы для задачи классификации с несколькими метками. Постановка задачи классификации с несколькими метками возникает во многих различных областях, например, при категоризации текста или тегировании изображений, и все они влекут за собой свои собственные особенности и проблемы. В обзоре [7] исследуются основы обучения с использованием нескольких меток, обсуждаются хорошо зарекомендовавшие себя методы, а также самые последние подходы. Возникающие тенденции рассматриваются в более свежем обзоре [1].

В работе [8] рассматривается та же постановка задачи классификации с несколькими метками в потоке событий, что и у нас. Модель авторов нацелена на фиксацию временных и вероятностных зависимостей между типами параллельных событий путем кодирования исторической информации с помощью энкодера, а затем использования условной смеси экспертов Бернулли. В этой статье [9] обсуждается постановка задачи прогнозирования временных наборов для пользователей, она предлагает систему непрерывного обучения, которая позволяет явно фиксировать изменяющиеся пользовательские предпочтения, поддерживая банк памяти, который мог бы хранить состояния всех пользователей и элементов. В этой парадигме авторы строят неубывающую универсальную последовательность, содержащую все пользовательские взаимодействия, а затем в хронологическом порядке извлекают уроки из каждого взаимодействия. Для исследования взаимосвязи между продуктами в корзине был предложен ConvTSP [10], который объединяет динамические интересы пользователей и статистические интересы в единое векторное представление.

Рекомендательные системы. В этом разделе мы представим статьи, связанные с проблемой рекомендации следующей корзины. Эта формулировка похожа на нашу, поэтому мы также рассмотрели многие подходы и идеи при анализе нашей области исследований. Авторы в [11] предложили персонализированную модель, которая фиксирует краткосрочные зависимости внутри временного набора продуктов, а также долгосрочную, основанную на исторической информации о пользователях. Также в [12] для соединения локальной и глобальной пользовательской информации предлагается гибридный метод, основанный на автоэнкодере для извлечения контекста и RNN для понимания динамики изменения интересов. Чтобы преодолеть подобные проблемы, для предсказания следующей рекомендации создается сеть внимания на основе графов, использующая hyper-edge подход [13]. При такой постановке задачи возникает сложность работы со словарем товарных категорий, поскольку они насчитывают тысячи значений, [14] использует GRU для прогнозирования следующей корзины, которая легко масштабируется до большого ассортимента.

Выявление взаимозависимости между метками с использованием алгоритма, основанного на собственном внимании в задаче классификации с несколькими метками.

Вклад. Мы разработали архитектуру на основе трансформера на основе собственного внимания между метками для работы над задачей классификации последовательностей событий по нескольким меткам. Наш основной вклад заключается в следующем:

- Мы вводим архитектуру LANET для прогнозирования набора меток для текущего события, используя информацию из предыдущих событий. Особенностью архитектуры является вычисление собственного внимания между представлениями меток.
- LANET превосходит модели на основе трансформера, которые фокусируются на вычислении собственного внимания между временными метками. Мы оцениваем все показатели в различных наборах данных(будет дополнение).

2 Постановка задачи

Мы рассмотрим классификацию с несколькими метками для последовательности $S = \{(X_i, Y_i)\}_{i=1}^{t-1}$. Он состоит из набора меток Y_i и набора признаков X_i, специфичных для каждой временной метки от 1 до $t-1$. Индекс соответствует времени события, поэтому (X_1, Y_1) - это информация о первом событии, а (X_{t-1}, Y_{t-1}) - это информация о последнем наблюдаемом событии. Множество $Y_i \subseteq \mathcal{Y}$, где $\mathcal{Y} = \{1, 2, \ldots, K\}$ - это множество всех возможных меток. Установленный размер X_i равен размеру Y_i. Каждая метка из Y_i сопровождается числовым или категориальным признаком из X_i в соответствующей позиции.

У нас также может быть дополнительный вектор признаков $\mathbf{z}$, описывающий рассматриваемую последовательность S в целом, например, идентификатор пользователя. Цель последовательной классификации с несколькими метками состоит в том, чтобы предсказать набор меток Y_t для следующей временной метки.

Мы создаем функцию $f(\cdot) \in [0,1]^K$, которая принимает историческую информацию о событиях в качестве входных данных и выводит вектор оценок для каждой из меток K. Эти оценки представляют собой вероятности присутствия метки в следующем наборе, связанных с событием.

В нашей настройке мы ограничиваем размер прошлого, доступного модели. $S^t = \{(X_j, Y_j)\}_{j=t-\tau}^{t-1}$, где τ означает количество событий, предшествующих рассматриваемому событию, с отметкой времени t, которая равна приписывается целевому набору меток Y_t. Итак, более формально $f(\cdot)$ имеет вид:

$$f(X_{t-\tau}, \ldots, X_{t-1}, Y_{t-\tau}, \ldots, Y_{t-1}, \mathrm{z}) \in [0,1]^K$$

для предсказания Y_t.

Чтобы завершить прогноз, нам нужна отдельная модель принятия решений о метках $g(f(\cdot))$, которая преобразует доверительные баллы в метки. Например, мы сравниваем оценку для k-й метки с выбранным пороговым значением β_k: если $f_k(\cdot) > \beta_k, k = 1, \ldots, K$, то модель предсказывает, что k-я метка присутствует. Таким образом, модель g создает набор меток $\hat{Y}_t \subseteq \mathcal{Y}$ на основе входных оценок достоверности.

Результирующее качество модели зависит как от метода $g(\cdot)$ для выбора меток для конечного набора, так и от производительности $f(\cdot)$ для получения достоверных оценок, в то время как мы сосредоточены на работе с $f(\cdot)$. Далее мы преобразуем задачу классификации с несколькими метками в задачи множественной бинарной классификации и оптимизируем модель, минимизируя потери перекрестной энтропии.

3 Архитектура LANET

Большинство моделей, связанных с трансформаторами, используемых для последовательного предсказания с несколькими метками, используют вычисление собственного внимания между последовательными представлениями входных временных меток. Вместо этого LANET использует собственное внимание между представлениями меток. Итак, у него есть входные данные, состоящие из K векторов. Ниже мы опишем, как агрегировать последовательность векторов размером от τ до K с помощью Слоя векторных представлений. Затем мы определяем Слой собственного внимания. Чтобы получить предсказания, мы применяем Слой предсказания.

Слой векторных представлений. Мы используем следующий подход для использования различных частей входных данных для последовательностей событий с несколькими метками:

- Векторное представление идентификаторов: Для идентификаторов мы изучаем матрицу эмбеддингов;
- Векторное представление времени: Для каждой временной метки мы знаем значение dt, которое равно разнице в днях между рассматриваемой и предыдущей временной меткой. Мы обучаем представления для каждого наблюдаемого значения dt. Мы также учитываем порядок событий, поэтому мы смотрим на предствления для позиций: $1, \ldots, \tau$, чтобы добавить их к dt представлению соответствующей временной метки;
- Представление суммы: Мы преобразуем все суммы в ячейки, разбивая непрерывные суммы сумм на интервалы. Каждому интервалу присваивается уникальный номер. Затем для каждого уникального номера мы создаем представление.
- Transformer label encoder: Для построения входных данных LANET мы используем данные, связанные с определенным идентификатором для временных меток τ. Мы объединяем представление меток, которые встречаются во время временных меток τ, с соответствующими векторами времени и суммы. Если метка не принадлежит истории последних временных меток τ, мы добавляем к ней векторы нулей в виде представлений времени и суммы. Если конкретная метка встречается несколько раз на предыдущих шагах τ, то мы создаем временные представления для каждого отдельного вхождения, а затем суммируем их, чтобы получить окончательное представление времени для этой метки. Мы делаем то же самое при построении вектора суммы в ситуации повторения метки.

Итак, в результате слоя вложения у нас есть $K+1$ векторов вложения. Первый вектор $\mathbf{e}_0$ соответствует вложениям общих (ID) признаков для последовательности. Все остальные векторы $\mathbf{e}_k, k = 1, \ldots, K$ являются объединением вложения метки с соответствующими вложениями времени и суммы. Оказывается, что векторы для исторически не задействованных меток — это просто вложения меток, поскольку мы суммируем их с нулевыми векторами времени и количества просмотров. При обучении всех весов встраивания они инициализируются из нормального распределения $\mathcal{N}(0, 1)$ и затем оптимизируются.

Собственное внимание. После получения представлений из наших данных мы переходим к компоновке нашей архитектуры. Пусть $E = \{\mathbf{e}_0, \ldots, \mathbf{e}_K\}$, $\mathbf{e}_i \in \mathbb{R}^d$ - последовательность входных представлений в энкодер, где $\mathbf{e}_0$ соответствует представлению $\mathbf{z}$ и все остальные $\mathbf{e}_i$ соответствуют представлениям, фиксирующим историческую информацию с точки зрения метки. В архитектуре Transformer влияние представление $\mathbf{e}_j$ на представление $\mathbf{e}_i$ достигается с помощью собственного внимания. Вес внимания α_{ij} и обновленный вектор $\mathbf{e}'_i$ рассчитываются как:

$$\mathbf{e}'_i = \sum_{j=0}^{K} \alpha_{ij}(W^v \mathbf{e}_j);\ \alpha_{ij} = \text{softmax}\left((W^q \mathbf{e}_i)^T (W^k \mathbf{e}_j)/\sqrt{d}\right),$$

где W^k — весовая матрица ключа, W^q - весовая матрица запроса, а W^v - весовая матрица значения. Такую процедуру внедрения обновлений можно повторить несколько раз, указав количество уровней преобразователя-энкодера.

Уровень прогнозирования. Обновленные представления обрабатываются одним полносвязным слоем, чтобы получить окончательные представления для каждой метки $\{\mathbf{e}_j^{(final)}\}_{j=1}^K$. Таким образом, мы получаем $\{f_j\}_{j=1}^K$, которые используются в классификаторе с несколькими метками с пороговым значением t_j, выбранным отдельно для каждой метки с использованием валидационной выборки.

4 Планирование эксперимента

4.1 Описание данных

Набор данных о продажах [15] - это исторические данные о продажах в разных магазинах. Метки относятся к категориям товаров, а сумма —- это количество проданных товаров для определенной категории.

Dataset	# events	Median set size	Max set size	# unique labels	Diff
Sales	47 217	16	48	84	0.0632
Demand	5 912	13	24	33	0.0957

Таблица 1: Характеристики наборов данных, используемых в задачах последовательной классификации с несколькими метками.

Набор данных о спросе [16] описывает исторический спрос на продукцию нескольких складов. Функция метки означает категорию продукта, а функция количества относится к соответствующему спросу.

Общая статистика по рассмотренным наборам данных приведена в таблице 1.

Мы представляем количество наблюдаемых событий, средний размер набора всех доступных наборов меток $\text{median}(|Y_{ij}|_{i,j=1,1}^{n,t_i})$, максимальный размер набора меток, который встречается в набор данных $\max(|Y_{ij}|_{i,j=1,1}^{n,t_i})$, количество уникальных меток K и Diff. Diff измеряет дисбаланс меток: мы вычисляем 5% и 95% квантилей для частот меток и берем разницу между ними. Представление меток не сбалансировано в большинстве наборов данных, но мы используем метрики, которые учитывают этот эффект.

5 Основной эксперимент

Эксперимент проводится на двух выборках Sales и Demand. Данные выборки представлены в таблице 1. Основным результатом будет сравнение подхода с собственным вниманием между представлениями на основе агрегации по меткам классов и подходом, использующим временную агрегацию для представлений. Также будут представлены базовые подходы, работающие с временными рядами это градиентный бустинг и LSTM. Сравнение представлено в таблице(тут будет таблица.).

Стоит изучить, какой подход к вычислению собственного внимания более важен: между метками или между временными метками. Таблица ... отвечает на этот вопрос. Label-attention - это наша базовая реализация. Time-attention - это случай, когда мы учитываем внимание только между просмотрами временных меток. Concat-attention подразумевает получение оценок достоверности путем объединения label-attention и time-attention. Мы узнаем коэффициенты важности двух форм внимания и используем их в качестве весов при суммировании просмотров внимания в случае gated-attention. Эксперимент с индикацией отсутствия заключается в добавлении обучаемого вектора к входным вложениям, если конкретная метка не участвует в рассматриваемой истории. тут какие-то выводы по таблице 2

Важным параметром нашей работы является ограничение исторической информации с помощью параметра τ. В нашей работе мы концентрируемся на событиях с временными метками. Модель учитывает только последние временные метки для прогнозирования. Таким образом, возникает естественный вопрос, как качество модели зависит от количества временных меток, используемых для построения прогноза. Длина входной последовательности для модели равна параметру look_back. Зависимость показателя micro-AUC от look_back представлена в 3

Список литературы

[1] Weiwei Liu, Haobo Wang, Xiaobo Shen, and Ivor Tsang. The emerging trends of multi-label learning. IEEE transactions on pattern analysis and machine intelligence, 2021.

[2] Thibaut Durand, Nazanin Mehrasa, and Greg Mori. Learning a deep convnet for multi-label classification with partial labels. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 647–657, 2019.

[3] Lin Xiao, Xin Huang, Boli Chen, and Liping Jing. Label-specific document representation for multi-label text classification. In Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP), pages 466–475, 2019.

[4] Adane Nega Tarekegn, Mario Giacobini, and Krzysztof Michalak. A review of methods for imbalanced multi-label classification. Pattern Recognition, 118:107965, 2021.

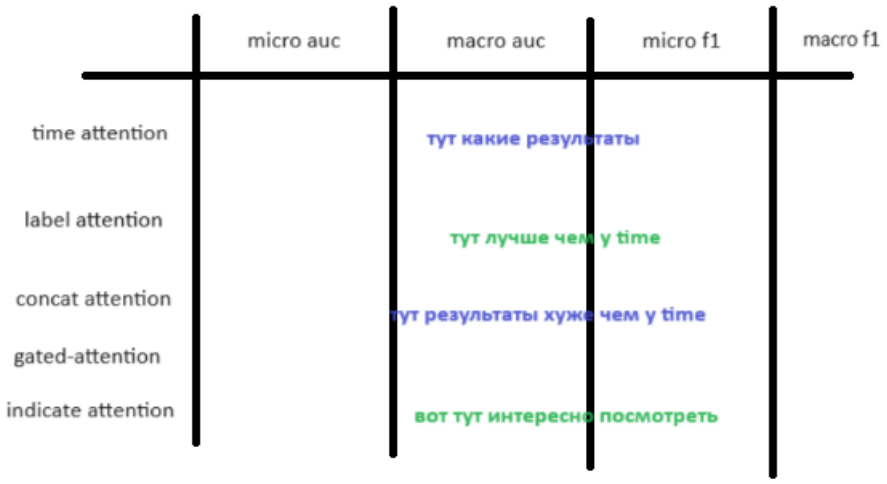

Рис. 2: Caption

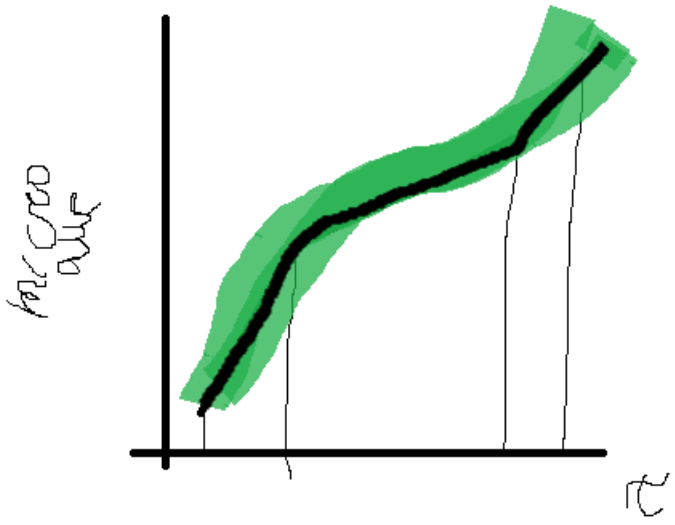

Рис. 3: Caption

[5] Thomas Hartvigsen, Cansu Sen, Xiangnan Kong, and Elke Rundensteiner. Recurrent halting chain for early multi-label classification. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pages 1382–1392, 2020.

[6] Wenyu Zhang, Devesh K Jha, Emil Laftchiev, and Daniel Nikovski. Multi-label prediction in time series data using deep neural networks. arXiv preprint arXiv:2001.10098, 2020.

[7] Min-Ling Zhang and Zhi-Hua Zhou. A review on multi-label learning algorithms. IEEE transactions on knowledge and data engineering, 26(8):1819–1837, 2013.

[8] Xiao Shou, Tian Gao, Shankar Subramaniam, Debarun Bhattacharjya, and Kristin Bennett. Concurrent multi-label prediction in event streams. In AAAI Conference on Artificial Intelligence, 2023.

[9] Le Yu, Zihang Liu, Leilei Sun, Bowen Du, Chuanren Liu, and Weifeng Lv. Continuous-time user preference modelling for temporal sets prediction. IEEE Transactions on Knowledge and Data Engineering, 2023.

[10] Fan Zhang, Shuai Wang, Yongjie Qin, and Hong Qu. Conv-based temporal sets prediction for next-basket recommendation. In 2023 International Conference on Frontiers of Robotics and Software Engineering (FRSE), pages 419–425. IEEE, 2023.

[11] Mozhdeh Ariannezhad, Ming Li, Sebastian Schelter, and Maarten de Rijke. A personalized neighborhood-based model for within-basket recommendation in grocery shopping. In Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, pages 87–95, 2023.

[12] V Ramanjaneyulu Yannam, Jitendra Kumar, Tejaswini Vankayala, and Korra Sathya Babu. Hybrid approach for next basket recommendation system. International Journal of Information Technology, 15(3):1733–1740, 2023.

[13] Tengshuo Song, Feng Guo, Haoran Jiang, Wenyun Ma, Zhenbao Feng, and Lei Guo. Hgat-br: Hyperedge-based graph attention network for basket recommendation. Applied Intelligence, 53(2):1435–1451, 2023.

[14] Luuk Van Maasakkers, Dennis Fok, and Bas Donkers. Next-basket prediction in a high-dimensional setting using gated recurrent units. Expert Systems with Applications, 212:118795, 2023.

[15] 1C Company. Predict Future Sales. `https://www.kaggle.com/c/competitive-data-science-predict-future-sales/data`, 2018.

[16] FELIXZHAO. Forecasts for Product Demand. `https://www.kaggle.com/datasets/felixzhao/productdemandforecasting`, 2017.