ДОКЛАДЫ РОССИЙСКОЙ АКАДЕМИИ НАУК. МАТЕМАТИКА, ИНФОРМАТИКА, ПРОЦЕССЫ УПРАВЛЕНИЯ, 2023, том 514, № 2, с. 118–125

УДК 517.54

ЭФФЕКТИВНОЕ ОБУЧЕНИЕ ГРАФОВЫХ СЕТЕЙ НА МНОГОМЕРНЫХ МНОГОСЛОЙНЫХ ПРЕДСТАВЛЕНИЯХ ТАБЛИЧНЫХ ДАННЫХ

© 2023 г. А. В. Медведев[1,*], А. Г. Дьяконов[2,]**

Представлено академиком РАН А.Л. Семеновым
Поступило 01.09.2023 г.
После доработки 15.09.2023 г.
Принято к публикации 18.10.2023 г.

Для задач предсказания на табличных данных дополнительная информация о целевой переменной может быть скрыта в отношениях между объектами. В частности если для таких объектов можно построить граф, где они будут вершинами, а связи между ними будут выражаться ребрами. Недавние работы показали, что совместное обучение графовых нейронных сетей и градиентных бустингов на таких данных дает прирост качества предсказания. В данной статье мы предлагаем новые методы обучения на табличных данных с графовой структурой. Эти методы являются попытками унифицировать современные многослойные модели для обработки табличных данных и графовые нейронные сети. Мы также предлагаем способы борьбы с вычислительной сложностью реализованных моделей и проводим наши эксперименты для индуктивных и трансдуктивных случаев. Наши результаты показывают, что предложенные модели обеспечивают качество, сравнимое с современными подходами.

Ключевые слова: табличные данные, графовые нейронные сети

DOI: 10.31857/S2686954323601628, **EDN:** GQQNLE

1. ВВЕДЕНИЕ

Под влиянием успеха глубоких нейронных сетей на данных со структурой в виде сетки (изображения) стали появляться работы, посвященные поиску операции свертки для данных с графовой структурой. Такие данные представлены не только признаковыми описаниями объектов X, но и графом G, вершинами которого являются объекты выборки. Такие новые типы сверток были успешно применены к задачам распознавания и предсказания позы человека [1], задачам классификации вершин графов цитирований и социальных сетей [2]. Они имеют потенциал в обработке категориальных признаков [3] и в задаче многоклассовой классификации [4].

Существует лишь несколько работ, посвященных обучению графовых нейронных сетей на табличных данных, где признаки, как правило, имеют очень разную природу и масштаб. Методы, основанные на решающих деревьях, обычно обладают наилучшим качеством обработки таких данных. Недавно появился целый ряд новых методов, которые позволяют обучать ансамбли и бустинги решающих деревьев с помощью методов распространения меток и обратного распространения ошибки. Эти новые модели позволяют получить многослойное представление табличных данных, которое, согласно нашему предположению, может быть эффективно использовано в качестве входных данных для графовых сверток. Более того, вышеупомянутые модели и графовые нейронные сети могут быть обучены совместно и, таким образом, быть более эффективными. Мы изучаем эти подходы и предлагаем модификации для повышения их качества и вычислительной эффективности. Наш основной вклад заключается в следующем:

- Мы обобщаем модель [7], делая бустинг многомерным. Затем мы изучаем влияние предобучения этого бустинга на общую производительность комбинированной модели. Мы также предлагаем процедуру предобучения для нашей обобщенной модели, которая позволяет ей быстрее сходиться к меньшим значениям функции потерь.

- Предлагаем дистилляцию многомерного бустинга, которая позволяет контролировать потребление памяти и время итераций обучения без снижения качества.

- Проводим эксперименты с совместным обучением графовой сверточной сети и ряда недавно предложенных дифференцируемых моделей на

[1]*ООО "Яндекс", Москва, Россия*
[2]*Центральный университет, Москва, Россия*
**E-mail: fortunato.mav@gmail.com*
***E-mail: djakonov@mail.ru*

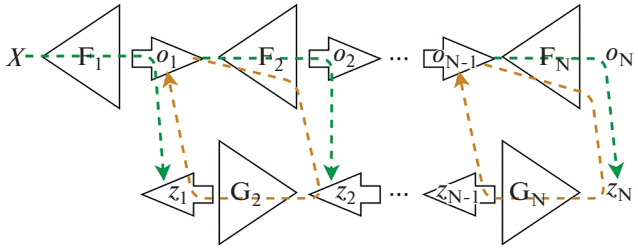

Рис. 1. Схема обучения MGBDT.

основе деревьев решений. Мы оцениваем полученные решения в задачах классификации вершин графа в индуктивных и трансдуктивных постановках.

2. ПРЕДЫДУЩИЕ РАБОТЫ

Работа [7] изучает проблему применения графовых нейронных сетей к разнородным табличным данным. Авторы совместно обучают градиентный бустинг (GBDT) и графовую нейронную сеть. Основная идея состоит в том, чтобы рассматривать графовую нейронную сеть как сложную функцию потерь для модели GBDT. Такой подход продемонстрировал значительный прирост качества и является нашим основным бейзлайном.

В работе [5] описано многослойное обобщение градиентного бустинга (MGBDT). Данная модель обучается не алгоритмом обратного распространения ошибки, а с помощью процедуры обратного распространения целевых переменных [11]. Данная процедура предполагает, что для каждого слоя обучается пара моделей (см. рис. 1). Модель F_i учится предсказывать входы для следующего слоя $i+1$ по выходам предшествующего ей $i-1$, модель G_i учит обратное F_i отображение.

Модель MGBDT позволяет получать эффективные для решения задач векторные представления на основе табличных данных, превосходя по качеству полносвязные нейронные сети. Однако это решение имеет ряд недостатков. Одним из которых является низкая эффективность использования памяти, поскольку каждый выход слоя фактически является отдельной моделью GBDT. Другой проблемой является увеличение времени обучения с количеством итераций. Для обычной модели GBDT мы можем сохранить ее выходные данные и использовать их для быстрого и эффективного обновления целевой переменной. Но когда мы рассматриваем многослойную модель, эта схема становится невозможной, поскольку входные данные для всех слоев, кроме самого первого, меняются с каждой итерацией.

Модель NODE, описанная в [6], обучается на табличных данных с помощью обратного распространения ошибки и таким образом лишена основных недостатков многослойного бустинга. Каждый слой такой модели представлен дифференцируемой версией ансамбля решающих деревьев. Обычное решающее дерево невозможно оптимизировать с помощью градиентного спуска, так как выборы признака, порога для каждого решающего правила не являются дифференцируемыми операциями. Были предложены аналоги таких операций на основе entmax [12] – разряженной версии softmax.

Широко известные модели глубинного обучения показывают отличные результаты на объектах, представляющих собой последовательности или простые сетки (например изображения). В то же время существует множество данных, представленных в виде графов. Главные отличия таких данных в том, что у каждого объекта-вершины может быть произвольное количество связанных с ним объектов-вершин, и на этом множестве нет никакого порядка. Примером таких данных могут служить социальные сети, статьи или посты с цитированиями и ссылками, молекулы. Вышеупомянутые особенности не позволяют с легкостью применить одну из основных операций – свертку.

Существует два основных подхода к введению операции графовой свертки:

• Спектральный подход – изначально был основан на спектральном разложении матрицы Лапласа графа [21]. Такое разложение служит аналогом разложения Фурье, из которого по теореме о свертке следует и определение такой операции на графе.

• Пространственные подходы вводят операцию свертки следующим образом:

$$x_v^t = \mathrm{COMBINE}^t(x_v^{t-1}, \mathrm{AGGREGATE}^t(\{(x_v^{t-1}, x_w^{t-1}) : (v,w) \in E\}),$$

здесь x_v^t – скрытое представление вершины v на слое t, $\mathrm{COMBINE}^t$, $\mathrm{AGGREGATE}^t$ – дифференцируемые операции, E – множество ребер графа.

В данной работе нас больше интересует второй подход, так как он позволяет проводить стохастическое обучение. В частности, в наших экспериментах мы используем модель GAT (Graph Attention Network) [14]. В этом случае указанные операции принимают следующий вид:

$$\mathrm{AGGREGATE}^t = \sum_{j \in \mathcal{N}_i} \alpha_{i,j} W x_j^{t-1},$$

$$\text{COMBINE}^t = \alpha_{i,i} W x_i^{t-1} + \text{AGGREGATE}^t,$$

здесь $\alpha_{i.j}$ – коэффициенты, полученные с помощью механизма self-attention [22]. Еще одно преимущество пространственного подхода – это возможность обучения по подграфам [13]. Операция сэмплирования окрестностей вершин является регуляризатором при индуктивном обучении, позволяющем модели лучше адаптироваться к меняющимся во времени графовым структурам.

3. ПРЕДЛОЖЕННЫЙ МЕТОД

В нашей работе мы пытаемся улучшить алгоритм, описанный в [7].Один из наших экспериментов (табл. 2, 3, 4) показал, что качество улучшается, когда графовая сеть принимает только выходы GBDT без исходных представлений. Мы предполагаем, что необработанные табличные данные (даже нормализованные) являются плохими входами для оператора свертки графа. С другой стороны, графовая модель, которая опирается только на приближение целевой переменной, немного наивна и может быть дополнена более сложным скрытыми представлениями вершин графа.

Мы сосредоточимся на использовании моделей [5, 6], чтобы обучить отображение табличных данных в пространство, в котором работает графовая свертка. Чтобы снизить вычислительную сложность и ускорить процесс обучения, мы предложили процедуры предобучения и дистилляции.

Один из важных аспектов модели BGNN [7] – это предобучение бустинга. Первые несколько итераций алгоритм учится предсказывать целевую переменную, затем выходы бустинга подаются на вход графовой сети и в дальнейшем обучении напрямую с целевой переменной никак не связаны. Для модели MGBDT [5] мы также изучаем влияние предобучения на качество финального предсказания. Однако в нашем алгоритме мы желаем отказаться от ограничения на размерность пространства, над которым действует графовая свертка. В таком случае мы сравниваем два варианта предобучения:

1. Добавление полносвязного слоя, который позволил бы модели учиться на целевую переменную.

2. Добавление к выходу размерности, которая приближает целевую переменную.

Ограничениями модели MGBDT, как уже говорилось, являются использование большого количества памяти и все время увеличивающееся время обучения. Наш подход, который мы называем дистилляцией, заключается в периодическом обновлении модели: новая модель инициализируется так, чтобы как можно лучше приближать предшествующую, при этом новая модель легче – в ней меньше регрессионных деревьев. Так как отображение из табличных данных постепенно обновляется, старые регрессионные деревья с первых итераций обучения уже не вносят ощутимый вклад в предсказание. Новые деревья исправляют ошибку старых регрессоров, адаптируя их выходы под постоянно меняющиеся представления вершин графа, над которыми работает графовая свертка.

Наш подход устраняет эту неэффективность, появляющуюся из-за постоянного изменения целевых переменных градиентного бустинга. Новая модель сходится к сравнимым результатам за меньшее время и без неограниченного потребления памяти.

Таблица 1. Основные характеристики наборов данных

	House	County	VK
Вершины	20640	3217	54028
Ребра	182146	12684	213644
Признаки	6	7	14

4. ЭКСПЕРИМЕНТЫ

Эксперименты проводятся на нескольких наборах данных[1]:

1. House [8] – набор данных, в котором каждая вершина – это недвижимость, ребра соединяют находящиеся рядом элементы недвижимости, целевая переменная – цена недвижимости.

2. County [9] – данные о выборных округах. Каждая вершина – это округ, ребрами соединены такие округи, которые имеют общую границу. Целевая переменная – уровень безработицы.

3. VK [10] – данные социальной сети. Целевая переменная – возраст пользователей.

В каждом нашем эксперименте мы оцениваем метрику качества – среднеквадратичную ошибку регрессии RMSE на кросс-валидации с помощью выборочных среднеквадратичного отклонения и матожидания, подсчитанных на 5 подвыборках каждого набора данных. Все графики мы строим на первой валидационной подвыборке.

Из табл. 1 следует, что характеристики графов рассмотренных наборов данных сильно различаются. Особенно сильно отличается распределение степеней вершин. Это может сильно отразиться на результатах обучения с сэмплированием подграфа.

Для начала мы изучаем влияние добавления ко входам графовой свертки необработанных таб-

[1] Выбор наборов данных обусловлен необходимостью сравнения предложенных в данной работе алгоритмов и алгоритмов, рассмотренных в работе [7].

Таблица 2. RMSE для различных графовых сверток на датасете [8]

GBDT-only	bgnn-agnn	bgnn-appnp	bgnn-gat	bgnn-gcn
True	0.49 ± 0.01	0.59 ± 0.01	0.49 ± 0.01	0.54 ± 0.01
False	0.53 ± 0.01	0.60 ± 0.01	0.55 ± 0.02	0.57 ± 0.01

Таблица 3. RMSE для различных графовых сверток на датасете [9]

GBDT-only	bgnn-agnn	bgnn-appnp	bgnn-gat	bgnn-gcn
True	6.87 ± 0.2	12.36 ± 0.15	7.00 ± 0.2	7.02 ± 0.2
False	6.96 ± 0.2	13.41 ± 0.16	6.95 ± 0.2	8.32 ± 0.43

Таблица 4. RMSE для различных графовых сверток на датасете [10]

GBDT-only	bgnn-agnn	bgnn-appnp	bgnn-gat	bgnn-gcn
True	1.27 ± 0.08	1.35 ± 0.13	1.26 ± 0.09	1.31 ± 0.11
False	1.29 ± 0.11	1.38 ± 0.14	1.33 ± 0.13	1.37 ± 0.07

личных данных. Мы использовали несколько видов графовых нейронных сетей, чтобы убедиться, что эффект не зависит от архитектуры: AGNN (графовая сеть с механизмом внимания) [16], APPNP (архитектура, берущая за основу персонализированный алгоритм случайного блуждания) [17], GAT (еще один подход на основе механизма внимания) [14], GCN (графовая сверточная сеть) [18].

Результаты можно увидеть в табл. 2, 3, 4. Очевидно, что графовая свертка не способна хорошо обрабатывать табличные данные и на самом деле лучшее качество получается на единственном признаке: предсказании модели, основанной на решающих деревьях.

Таблица 5 и рис. 2 показывают, что предобучение оказывает большое влияние на сходимость обеих моделей. Результаты обеих процедур предобучения продемонстрировали сопоставимый эффект, хотя предобучение одного из выходных параметров в среднем имеет несколько меньшее значение функции потерь в среднем. Мы предполагаем, что графовой модели может потребоваться больше информации о соседних вершинах графа, чем просто целевая метка, и эта информация фиксируется остальными выходами. Предобучение с помощью дополнительного линейного слоя накладывает ограничение на выходы модели GBDT: они фиксируют только информацию о целевых метках.

Таблица 6 содержит значения RMSE как для дистиллированной, так и для обычной версий модели MGBDT. Дистилляция происходит каждые 10 эпох. Рисунок 3 показывает, что процедура дистилляции практически не влияет на вид кривой функции потерь.

Рисунок 4 демонстрирует эффективность процесса дистилляции. Можно заметить, что время на одну итерацию остается ограниченным для дистиллированного варианта, в то время как в изначальном варианте оно неограниченно растет. Кроме того, процедура помогает держать потребление памяти под контролем.

5. РЕЗУЛЬТАТЫ

В данной работе мы проводим эксперименты для двух сценариев практического применения:

1. Индуктивная постановка предполагает, что графовые структуры на обучающей и тестовой выборке отличны. Это значит, что во время обучения в операции графовой свертки участвует не

Таблица 5. RMSE для различных стратегий предобучения MGBDT

	County	House	VK
None	1.285 ± 0.107	0.548 ± 0.019	6.970 ± 0.194
Linear	1.267 ± 0.087	0.521 ± 0.008	7.004 ± 0.183
Stack	**1.249 ± 0.081**	**0.518 ± 0.008**	**6.915 ± 0.227**

Таблица 6. RMSE для MGBDT с дистилляцией и без. Датасет House

	Размерность		
Дистилляция	4	8	16
True	0.559 ± 0.017	0.538 ± 0.004	0.529 ± 0.007
False	0.533 ± 0.011	0.522 ± 0.005	–

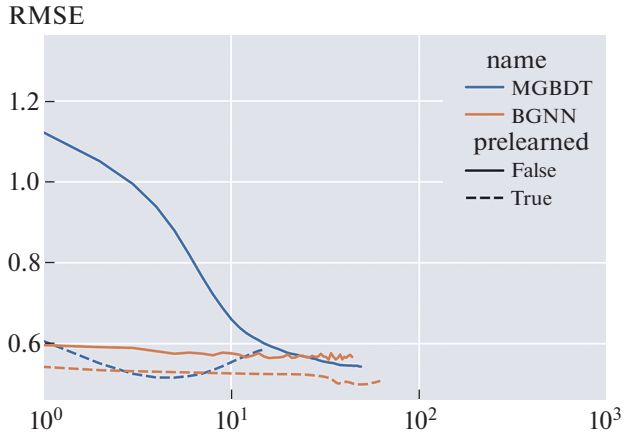

Рис. 2. Функция потерь от числа итераций (логарифмическая шкала) для MGBDT и BGNN.

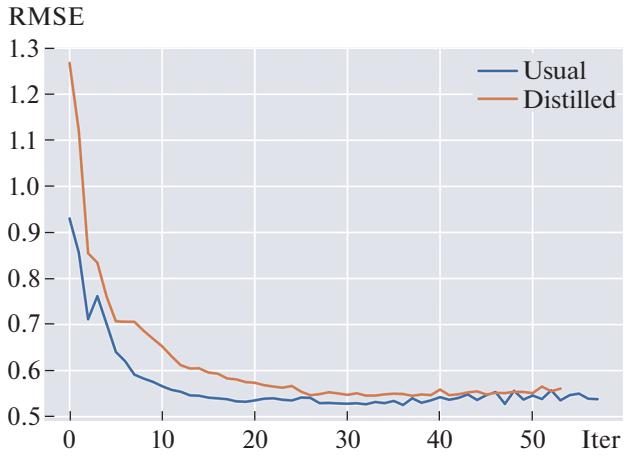

Рис. 3. Функция потерь для MGBDT.

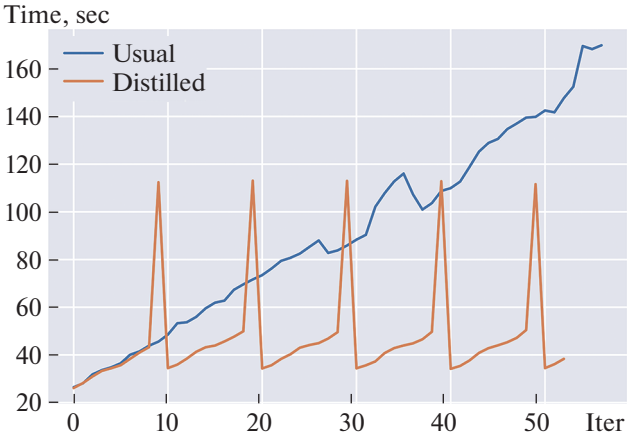

Рис. 4. Время на одну итерацию MGBDT.

вся окрестность вершины, а только те вершины, которые присутствуют в тренировочной выборке.

2. Трансдуктивная, наоборот, предполагает наличие общей структуры, для обеих выборок. То есть во время обучения модель использует все вершины графа в операции свертки.

В индуктивной постановке мы также рассматриваем обучение с регуляризацией – сэмплированием подграфов. Модель с NODE обучается таким алгоритмом во всех экспериментах, так как иначе требуется слишком много вычислительных ресурсов.

Для каждой модели мы подбираем гиперпараметры по сетке:

- размерности: [4, 8]
- количество слоев: [1, 2]
- вероятность зануления весов: [0, 0.2]

Глубина[2] деревьев для всех моделей равна 6, для модели NODE число деревьев на каждом слое берем равным 128, а размерность листового вектора 3.

Решения на основе градиентного бустинга в целом показали сравнимые результаты. К сожалению обучение модели NODE слишком ресурсоемко, и мы ограничили ее обучение 200 эпох вместо 5000 как для полносвязной сети. По характеру сходимости кривой обучения (рис. 4) алгоритм ведет себя скорее как полносвязная нейросеть, чем метод, основанный на решающих деревьях. Однако можно заметить, что даже с учетом того что модель NODE учится на подграфах, она сходится гораздо быстрее, чем другие полностью дифференцируемые методы (GAT, FCNN, FCNN-GNN).

По итогам обоих экспериментов, среди MGBDT моделей, лучше всего себя показали однослойные. Однако результаты из табл. 6 говорят, что многослойные модели потенциально могут показать еще лучшее качество при увеличении размерности слоев. Важно заметить, что каждый слой в MGBDT, кроме первого, имеет ограничение в виде необходимости учить обратимое отображение. Таким образом и можно объяснить худший по сравнению с однослойными моделями результат. При усложнении таких отображений, увеличении размерностей слоев, качество работы модели улучшается, а дистилляция помогает разумно ограничить необходимые вычислительные ресурсы.

На рис. 6 цветом показаны значения целевой переменной и того выхода скрытого слоя моделей MGBDT и BGNN, по которому шло предобучение. Хотя мы увеличили размеры выходного слоя и никак не ограничивали значения предобученной размерности в дальнейшем процессе настройки модели, тем не менее, как и в работе [7], предобученная размерность может быть интер-

[2] Хотя в разных алгоритмах используются разные типы решающих деревьев, для наших данных с небольшим числом признаков, на практике оптимальная глубина для каждого типа оказалась одинаковой.

Таблица 7. Результаты для случая трансдуктивного обучения

	House	County	VK
CatBoost	0.63 ± 0.01	1.39 ± 0.07	7.16 ± 0.2
LightGBM	0.63 ± 0.01	1.4 ± 0.07	7.2 ± 0.21
GAT	0.54 ± 0.01	1.45 ± 0.06	7.22 ± 0.19
FCNN	0.68 ± 0.02	1.48 ± 0.07	7.29 ± 0.21
FCNN-GNN	0.53 ± 0.01	1.39 ± 0.06	7.22 ± 0.20
BGNN	**0.5 ± 0.01**	**1.26 ± 0.08**	**6.95 ± 0.21**
MGBDT+GAT	**0.5 ± 0.01**	**1.25 ± 0.08**	**6.92 ± 0.23**
NODE+GAT (200 эпох)	0.6 ± 0.07	1.49 ± 0.1	–

Таблица 8. Результаты для случая индуктивного обучения

	House	County	VK
BGNN	0.509 ± 0.008	**1.254 ± 0.93**	**7.049 ± 0.222**
MGBDT+GAT	**0.503 ± 0.012**	1.264 ± 0.08	7.072 ± 0.19
BGNN reg	0.521 ± 0.006	1.312 ± 0.1	7.178 ± 0.171
MGBDT+GAT reg	0.520 ± 0.01	1.264 ± 0.08	7.275 ± 0.24

претирована как сглаженная версия целевой переменной. Координаты (x, y) в каждом случае равны результату работы алгоритма t-SNE [15], над выходами предпоследнего слоя графовой сети.

6. ЗАКЛЮЧЕНИЕ

В этой работе мы предложили новые модели для обучения на табличных данных. Наши решения используют преимущества совместного обучения решающих деревьев и графовых нейронных сетей. В отличие от [7], мы используем многослойные представления табличных данных высокой размерности и предлагаем эффективные с точки зрения вычислений способы их получения. Эксперименты на [8–10] показали, что наши алгоритмы достигают результатов, сопоставимых с последними современными моделями. Мы хотим провести дальнейшие исследования в направлении сокращения вычислительных ресур-

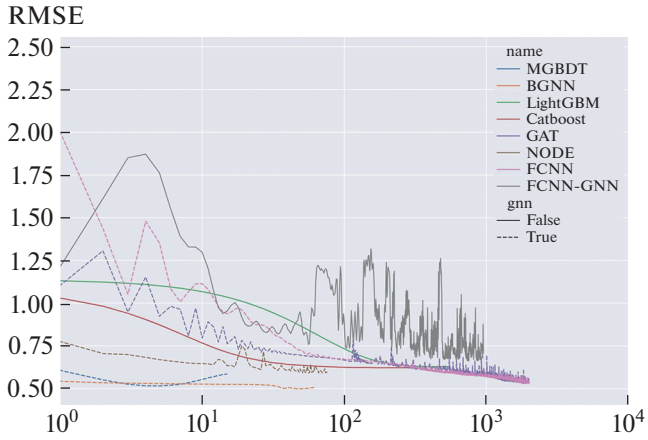

Рис. 5. Функция потерь от числа итераций (лог. шкала) каждой модели на датасете House [8].

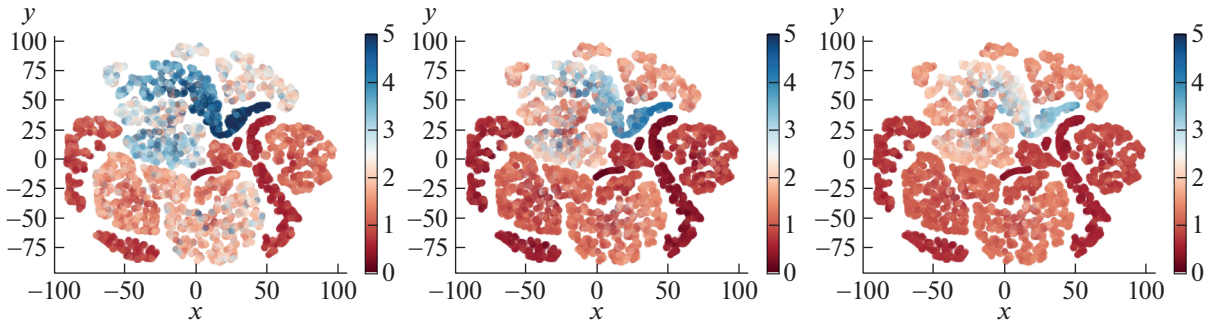

Рис. 6. Цветом показаны целевая переменная, MGBDT, BGNN по предобученной размерности.

сов путем дистиллирования решающих деревьев с помощью полносвязных сетей. Это помогло бы контролировать потребление памяти без периодической реинициализации. Мы также считаем, что наш подход более перспективен в отношении трансферного обучения, поскольку его размерность не сильно зависит от целевой переменной для текущей задачи, но необходимы дальнейшие исследования, чтобы доказать это.

СПИСОК ЛИТЕРАТУРЫ

1. *Li M., Chen S., Chen X., Zhang Y., Wang Y., Tian Q.* Actional-structural graph convolutional networks for skeleton-based action recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019. P. 3595–3603.
2. *Hamilton W.L., Ying R., Leskovec J.* Inductive representation learning on large graphs. 2017. arXiv preprint arXiv:1706.02216
3. *Li Z., Cui Z., Wu S., Zhang X., Wang L.* Fi-gnn: Modeling feature interactions via graph neural networks for ctr prediction. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management. 2019. P. 539–548.
4. *Chen Z.M., Wei X.S., Wang P., Guo Y.* Multi-label image recognition with graph convolutional networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019. P. 5177–5186.
5. *Feng J., Yu Y., Zhou Z.H.* Multi-layered gradient boosting decision trees. 2018. arXiv preprint arXiv:1806.00007
6. *Popov S., Morozov S., Babenko A.* Neural oblivious decision ensembles for deep learning on tabular data. 2019. arXiv preprint arXiv:1909.06312
7. *Ivanov S., Prokhorenkova L.* Boost then Convolve: Gradient Boosting Meets Graph Neural Networks. 2021. arXiv preprint arXiv:2101.08543
8. *Pace R.K., Barry R.* Sparse spatial autoregressions. Statistics & Probability Letters. 1997. V. 33. № 3. P. 291–297.
9. *Tsitsulin A., Mottin D., Karras P., Müller E.* 1Verse: Versatile graph embeddings from similarity measures. In Proceedings of the 2018 world wide web conference. 2018. P. 539–548.
10. *Jia Junteng, Austin Benson.* “Outcome correlation in graph neural network regression”. 2020. arXiv preprint arXiv:2002.08274
11. *Lee D.H., Zhang S., Fischer A., Bengio Y.* Difference target propagation. In Joint european conference on machine learning and knowledge discovery in databases. Springer, Cham, 2015. P. 498–515.
12. *Peters B., Niculae V., Martins A.F.* Sparse sequence-to-sequence models. 2019. arXiv preprint arXiv:1905.05702
13. *Hamilton W.L., Ying R., Leskovec J.* Inductive representation learning on large graphs. 2017. arXiv preprint arXiv:1706.02216
14. *Veličković P., Cucurull G., Casanova A., Romero A., Lio P., Bengio Y.* Graph attention networks. 2017. arXiv preprint arXiv:1710.10903
15. *Van der Maaten L., Hinton G.* Visualizing data using t-SNE. Journal of machine learning research. 2008. V. 9. № 11.
16. *Thekumparampil K.K., Wang C., Oh S., Li L.J.* Attention-based graph neural network for semi-supervised learning. 2018. arXiv preprint arXiv:1803.03735
17. *Klicpera J., Bojchevski A., Günnemann S.* Predict then propagate: Graph neural networks meet personalized pagerank. 2018. arXiv preprint arXiv:1810.05997
18. *Kipf T.N., Welling M.* Semi-supervised classification with graph convolutional networks. 2016. arXiv preprint arXiv:1609.02907
19. *Perozzi B., Al-Rfou R., Skiena S.* Deepwalk: Online learning of social representations. In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. 2014. P. 701–710.
20. *Grover A., Leskovec J.* node2vec: Scalable feature learning for networks. In Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. 2016. P. 855–864.
21. *Williamson C.* Spectral graph theory, expanders, and ramanujan graphs. University of Washington. 2014.
22. *Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A.N., Kaiser Ł., Polosukhin I.* Attention is all you need. Advances in neural information processing systems. 2017. V. 30.

TOWARDS EFFICIENT LEARNING OF GNN ON HIGH-DIMENSIONAL MULTI-LAYERED REPRESENTATIONS OF TABULAR DATA

A. V. Medvedev[a] and A. G. Djakonov[b]

[a] *"Yandex Company", Moscow, Russian Federation*

[a] *Centural University, Moscow, Russian Federation*

Presented by Academician of the RAS A.L. Semenov

For prediction tasks using tabular data, it is possible to extract additional information about the target variable by examining the relationships between the objects. Specifically, if it is possible to receive a graph in which the objects are represented as vertices and the relationships are expressed as edges, then it is likely that the graph structure will contain valuable information. Recent research has indicated that jointly training graph neural networks and gradient boostings on this type of data can increase the accuracy of predictions. This article proposes new methods for learning on tabular data that incorporates a graph structure, in an attempt to combine modern multilayer techniques for processing tabular data and graph neural networks. In addition, we discuss ways to mitigate the computational complexity of the proposed models, and we conduct experiments in both inductive and transductive settings. Our findings demonstrate that the proposed approaches provide comparable quality to modern methods.

Keywords: tabular data, graph neural networks