УДК 004.91:7.067

ОЦЕНКА ПАССАЖИРОПОТОКА И ТРАНСПОРТНОГО ТРАФИКА В ГОРОДСКИХ УСЛОВИЯХ

Л.В. Рудикова-Фронхёфер, Н.И. Игнатенко

Гродненский государственный университет имени Янки Купалы,
ул. Ожешко, 22, 230023, г. Гродно, Беларусь
lada.rudikowa@gmail.com, nikolay.ignatenko@outlook.com

В статье приведены подходы к экстраполяции транспортного трафика в условиях городской среды с использованием машинного обучения, в частности, метода ближайших соседей, градиентного бустинга, графовых и классических нейронных сетей. Модели разрабатывались для пассажирских посадок такси и пассажиропотока метрополитена.

Ключевые слова: машинное обучение; графовые нейронные сети; экстраполяция; трафик; анализ данных.

ESTIMATING OF PASSENGER AND TRANSPORT TRAFFIC IN URBAN CONDITIONS

L.V. Rudikova-Fronhoefer, N.I. Ihnatsenka

Yanka Kupala State University of Grodno,
Ozheshko str., 22, 230023, Grodno, Belarus,
lada.rudikowa@gmail.com, nikolay.ignatenko@outlook.com

The article presents approaches to extrapolation of transport traffic in a megalopolis using machine learning, in particular the nearest neighbor method, gradient boosting, graph and classical neural networks. The models were developed for passenger taxi landings and metro passenger traffic.

Keywords: machine learning; graph neural networks; extrapolation; traffic; data analysis.

Введение

Современный мир наблюдает рекордный рост и развитие городов в различных аспектах наравне с большим уровнем урбанизации. Город и его механизмы становятся более комплексными, в силу чего требуются наиболее визуальные и аналитические методы для отображения городских процессов, наблюдаемых на городских территориях. В области урбанистики активно применяется машинное обучение, а также достижения и результаты исследований из других различных сфер науки и техники. Эмпирически оказалось, что структура данных оказывает

ключевое значение для методологии машинного обучения, а появление графовых нейронных сетей влияет на получение корректных результатов в задачах с нерегулярной структурой данных.

Предметом предлагаемого исследования послужили данные посадок пассажиров такси, данные турникетов метрополитена, а также социо-экономические данные о локациях (например, численность населения, его занятость и т.п.) города Нью-Йорка.

Целью исследования являлось изучение регрессионных моделей в задаче экстраполяции, использующих наработки современных подходов к созданию графовых нейронных сетей, таких, как Graph Attention Networks (GAT) [1], GCN [2] и других методов машинного обучения, применимых к графам. Другой частью исследования было сопоставление выразительности моделей, использующих подход Message Passing в сравнении с традиционными детерминированными методами.

Среди работ, успешно применяющих графовые нейронные сети, следует отметить ETA (estimated time arrival) prediction в Google Maps [3]. Так ученые из DeepMind смогли улучшить показатели предыдущих систем на 16-51 % в зависимости от города. Ученые из Китая также вполне успешно использовали комбинацию сверточных и рекуррентных нейронных сетей совместно с методами обучения без учителя с целью получения векторных представлений вершин графа для предсказания скорости автомобильного потока [4].

1. Методология исследования / теоретические основы

Оценка трафика в локации, которая является ненаблюдаемой, является основной задачей работы, вследствие чего определим понятие трафика. В случае такси трафик в локации – это количество посадок в заданном радиусе за определенный временной интервал, а дорожная сеть задает непосредственно граф (вершины – перекрёстки либо часть дороги, ребра – дороги). Для данных метрополитена трафик задавался количеством прохождений через турникеты на станции. Данные такси насчитывали около 20000 вершин и 60000 ребер, метрополитена – около 350 вершин и 1500 ребер.

Поставим задачу экстраполяции трафика следующим образом: для части локаций наблюдается трафик (мы знаем его оценку), а для любой ненаблюдаемой локации, на основании ее признаков и наблюдаемого трафика других локаций, необходимо его оценить. В качестве базовой модели, относительно которой можно делать выводы о значимости последующих и более сложных моделей, использовался метод k-ближайших соседей. В данном случае требовалось определить

оптимальное число соседей, для чего использовалась библиотека для поиска оптимальных гиперпараметров. Оценкой трафика было среднее его значение по соседям. За метрику похожести объектов бралось географическое расстояние. Впоследствии, этот метод был модифицирован: определим вес ребра между локациями как w(a,b) и оценим трафик в локации как уже нормализированное взвешенное среднее (1):

$$T(a) = \sum_{a \neq b} \frac{w(a,b)}{\sum_{a \neq b} w(a,b)} T(b)\,, \qquad (1)$$

где T(a) – трафик в локации. В качестве функции w(a,b), которую можно охарактеризовать как релевантность между локациями (использовались несколько альтернатив (2), (3), (4)).

$$w(a,b) = exp\hat{}(-k * dist(a,b))\,, \qquad (2)$$

где параметр k здесь обучаемый, a dist (a, b) – географическое расстояние между локациями.

$$w(a,b) = \cos\_\mathrm{sim}(\mathrm{x(a)}, \mathrm{x(b)})\,, \qquad (3)$$

где x(a) – векторное представление вершины, которые были получены посредством подхода node2vec [5], cos_sim – косинус между векторными представлениями вершин. Node2vec – это алгоритм для генерации векторных представлений узлов на графе. Инфраструктура node2vec изучает низкоразмерные представления для узлов в графе с помощью случайных блужданий по графу, начиная с целевого узла. Важной особенностью этого подхода, является тот факт, что векторные представления узлов, находящихся структурно рядом, будут схожи, что выражается непосредственно как косинус между их векторными представлениями. Кроме cos_sim использовались и другие методы, но значительных улучшений в результате это не дало.

$$w(a,b) = MLP(x(a)||x(b))\,, \qquad (4)$$

где MLP – многослойный персептрон, || – операция конкатенирования векторов. Пробовались различные вариации конфигурации этой модели: векторные представления размером 64, 128 и 256 в тренировке node2vec (как и вышеперечисленных подходах), произведение Адамара вместо конкатенации, изменение структуры MLP (5).

$$(a,b) = \exp\left(LeakyReLU(v^T[Wx(a)||Wx(b)])\right) \qquad (5)$$

где для LeakyReLU negative slope был 0.2, v – параметризованный вектор, W – линейное преобразование.

По сути, данное выражение задаёт механизм «внимания», аналогичный приведенному в статье [1]. Как и в приведенной статье использовались несколько «головок» для получения более стабильных моделей.

Наилучшим вариантом для релевантных весов с случае посадок такси оказалась функция (2), поскольку при своей простоте она значительно не уступает другим более комплексным методам. Для увеличения производительности тренировки и оптимального расхода памяти для обучения использовалась лишь приблизительно шестая часть всей сети, более точно – центральная часть города в радиусе 8 км. На основание испытанных моделей, можно сделать следующий вывод: наиболее успешно описывает оценку трафика в локации наблюдаемый трафик в близлежащих точках. Этот результат, на наш взгляд, можно объяснить во многом следующей причиной: высокая плотность покрытия города дает достаточно большую корреляцию между соседними локациями, вследствие чего следует ожидать ухудшения показателей модели при более разреженном покрытии города. С этим и столкнулись, работая с данными станций метро.

Для метрополитена вышеперечисленные подходы, практически, не работают: R2 держится около нуля. Эти результаты во многом объясняются особенностями данных: средний трафик за день для двух соседних станций метро может разительно отличаться. Для решения этой проблемы для модели были определены коэффициенты масштабирования, которые должны показывать отношение значений целевой переменной локаций. Таким образом, вводим и обучаем функцию масштабирования s для локации от различных ее социо-экономических показателей (численность населения, его занятость и пр.), которая должна удовлетворять следующему выражению (6) и трафик будем оценивать по уравнению (7).

$$\frac{T(a)}{T(b)} \approx \frac{s(a)}{s(b)} \tag{6}$$

$$T(a) = s(a) \sum_{a \neq b} \frac{w(a,b)}{\sum_{b \in N(a)} w(a,b)} \frac{T(b)}{s(b)} \tag{7}$$

Следует отметит, что можно обучать функцию масштабирования независимо от экстраполяции. Так, обучая MLP вместе со всей моделью, не было получено значимых результатов. Более того, в процессе обучения функции масштабирования отдельно от самой модели, MLP не смог выйти на положительные значения R2. Заметим, аппроксимируя функцию масштабирования, решается следующая задача: по локальным признакам прогнозируется трафик для заданного дня. Наиболее хорошие результаты

в этом случае показал градиентный бустинг (в реализации использовался Catboost): 0.2 по R2. По сути, это говорит о том, что использованных признаков недостаточно, чтобы хорошо выразить зависимость между ними и наблюдаемым трафиком. Так, найдя подходящие признаки и построив точные модели для масштабирования локаций, можно улучшить основную модель. Чтобы искусственно обойти эту проблему использовался средний трафик за все время наблюдений для каждой станции в качестве функции масштабирования, что дало положительные результаты, свидетельствующие о рациональности подхода. Также отметим, что коэффициенты масштабирования должны зависеть как минимум от дня недели из предположения от изменчивости трафика в рабочие и выходные дни. Для этого можно использовать либо день недели как категориальный признак для функции масштабирования, либо для каждого дня недели строить собственную модель.

2. Результаты и их обсуждение

В последующей таблице приведены результаты тестирования моделей. Значение метрики вычислялось в среднем для месяца.

Таблица – Результаты тестирования моделей (R2)

Подход / Данные	kNN	HBC (2)	HBC (3)	HBC (4)	HBC (5)
Такси	0.75	0.88	0.8	0.82	0.86
Метро (с коэффициентами масштабирования)	0.7	0.77	0.76	0.74	0.79

Весь код написан на языке Python с использованием фреймворков для машинного обучения: PyTorch, PyTorch Geometric, Sklearn, Numpy. Для эффективной обработки данных применялись библиотеки Pandas и NetworkX. Все обучение проходило на видеокарте NVIDIA GTX 1050M (4 Gb).

Заключение

Таким образом, предлагаемая разработка предлагает путь к экстраполяции и улучшению качества анализа данных широкого спектра городских сенсорных наблюдений, таких, как качество воздуха. Результаты предиктивной модели могут быть использованы для

улучшения логистики и/или обнаружения аномальных сценариев (путем сравнения фактических наблюдений с прогнозами), чтобы информировать аналитиков о возможных аномалиях, к которым необходимо подготовиться. Кроме этого, работа показывает практическое применение новейших практик и методологий в машинном обучении на городских временных данных.

Библиографические ссылки

1. Veličković P. Graph Attention Networks // ArXiv:1710.10903 [Cs, Stat] [Электронный ресурс]. Feb. 2018. arXiv.org. URL: http://arxiv.org/abs/1710.10903. (дата обращения: 24.08.2022.)
2. Kipf Th.N. Semi-Supervised Classification with Graph Convolutional Networks // ArXiv:1609.02907 [Cs, Stat], 4 [Электронный ресурс]. Feb. 2017. arXiv.org. URL: http://arxiv.org/abs/1609.02907. (дата обращения: 26.08.2022.)
3. Derrow-Pinion A., et al. ETA Prediction with Graph Neural Networks in Google Maps // Proceedings of the 30th ACM International Conference on Information & Knowledge Management. Oct. 2021. P. 3767–3776.
4. Yu B., et al. Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting // Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence. July 2018. P. 3634–3640. doi.org/10.24963/ijcai.2018/505.
5. Grover A., Leskovec J. Node2vec: Scalable Feature Learning for Networks // ArXiv:1607.00653 [Cs, Stat] [Электронный ресурс]. July 2016. arXiv.org. URL: http://arxiv.org/abs/1607.00653 (дата обращения: 24.08.2022.)