FORECASTING SPARSE TRAFFIC CONGESTION PATTERNS USING MESSAGE-PASSING RNNS

Shiva R. Iyer[1]*, Ulzee An*[1]*, Lakshminarayanan Subramanian*

[1]Both authors contributed equally

Department of Computer Science, New York University, New York, USA

ABSTRACT

The ability to forecast traffic congestion ahead of time given road conditions has remained a prominent problem in road traffic analysis. In this work, we leverage mobility traces of public transport vehicles tracked by the New York City MTA and formulate Message-Passing Recurrent Neural Nets (MPRNN) to produce long-term traffic forecasting on data that is sparse but wide in coverage. We model the interactions among road segments spread over the entirety of Manhattan, New York over a period of 3 months, such that traffic conditions can be propagated to $> 90\%$ of examined segments from just a few observations. In comparison to other competing algorithms, MPRNN achieves the lowest mean error of < 0.3 mph when predicting ahead in 10 minute intervals, for up to 3 road segments ahead (message passing across 3 hops). The MPRNN model further offers compelling results when forecasting traffic speeds several hours ahead given distant observations up to approximately 1 kilometer away (three consecutive bus stops) with a mean error of about 2 mph.

***Index Terms*—** Road traffic, deep learning, message passing, recurrent neural networks

1. INTRODUCTION

In this work, we present a new take on the problem of traffic forecasting from sparse but easily available and accessible data such as mobility traces of public transportation vehicles. Many popular traffic prediction applications that are used for travel time prediction today such as Google Maps and Waze rely on large amounts of crowdsourced data from human mobile phone users on the road [1]. This crowdsourced approach is typically not useful or reliable in all but the big cities [2], since data is either unavailable or stale. Such an approach is also difficult to implement in regions with strict regulations on free data access and privacy concern among the public.

We demonstrate in this work that it is possible to forecast road traffic conditions, in particular the level of congestion, from sparse data collected from significantly fewer input sources. Our input sources are public transportation buses fitted with location trackers. A significant difference from more traditional takes in this problem is the *sparsity* of the data in time and space. On the space front, we note that buses typically ply only on select road segments in a city in a predictable and repeatable fashion, covering only a fraction of all the roads in a city. On the time front, we note that many related works use either taxi traces [3, 4, 5] or data from specialized instrumentation called loop detectors [6, 7, 8]. These data sources supply data nearly continually throughout the day by very nature of design, whereas buses do not ply with as much temporal frequency and spatial coverage at late nights as much as during the day times.

Accurate modeling of traffic flow and congestion requires additional features apart from vehicle traces that directly impact traffic flow, such as the number of lanes, frequency of stop lights and pedestrian density. However, the public transit authorities may either not collect such data or may not make them available to us. Without the full set of features, forecasting road conditions strays further from simulating a closed system and thus traffic patterns appear highly non-linear. In this paper, we motivate Message-Passing RNN (*MPRNN*) which reduces confounding effects through spatial awareness and models interactions between road segments such that forecasts are resilient to unreliable local measurements. Using the MPRNN, we are able to make longer-term forecasts of traffic speeds in both space and time using only limited input data from a small number of road segments.

Our contributions in this work are three-fold from a performance point-of-view. First is the novel application of the message-passing neural network formulation in the context of traffic congestion forecasting and mapping. We demonstrate improved prediction performance, as well as better and faster modeling of spatial interactions using the MPRNN formulation. Second, we show, for the first time, competitive forecasting results over a working day period (approx 12 hours). The MPRNN is able to predict next step traffic speeds with an impressively low error less than $0.3\,\mathrm{mph}$, and forecast over longer periods with an minimum error of about $1.8\,\mathrm{mph}$. Third, we demonstrate the ability to forecast speeds at road segments that are not immediately adjacent to observed road segments ("spatial" forecasting). In fact, we are able to forecast speeds in a segment using limited data from segments up to about one kilometer (0.6 mile) away.

2. RELATED WORKS

The problem of short-term road traffic forecasting is an area replete with studies, as summarized in an excellent and long review of the area [9], with discussions and references to more than 200 works. [10] is also another good review. Traditional approaches to travel time prediction such as ARIMA models and their variants, and Kalman filters, work well to estimate next-step future values in time series, and have been used with some moderate success in short-term traffic flow prediction [11, 12]. But as more recent works have repeatedly shown [6, 3, 8, 4, 7], ARIMA and similar approaches do not model spatial dependencies between connecting road links sufficiently, and thus not do not yield the best predictive performance. The congestion state in a road segment depends strongly on the states upstream as well as downstream. Second, ARIMA methods are extremely poor at long-term forecasting. And third, they assume that the time series data is stationary, which cannot be expected of the real traffic speeds. The most recent works cited here attempt to address many of these issues using different deep neural net architectures, but evaluated on datasets of a different nature viz. taxi traces and loop detector data, that are denser and richer than our dataset. We utilize a simpler architecture that works with sparse datasets such as ours, and yielding performance comparable to [6].

3. DATASET AND GRAPHICAL REPRESENTATION

The Metropolitan Transportation Authority (MTA) in New York City provides a raw datalog of locations reported continually by the MTA buses. The available information is not only sparse in space, but coarse. Each bus reports a timestamp, distance traveled in the trip, and a *stop code* referring to the next bus stop. We define a *segment* as the portion of a bus route between two consecutive bus stops. Data entries are received at an arbitrary interval close to < 1 minute, sometimes observed as low as 30 seconds. We utilized a downloadable historical dump [13] of all bus locations over a continuous period of 90 days in 2014 in Manhattan. The data contains information for about 42 different bus routes, covering over 685 bus stops, across the borough of Manhattan.

In the data preparation step, speeds are computed for each individual bus from the distance and timestamp information. The speeds from multiple buses are then aggregated at each segment at 10 minute intervals to create a time series for each segment. As a result of this procedure, traffic speed data is obtained at far greater spatial coverage than what would be obtained from loop detectors, which are usually placed only on arterial and peripheral roads. A **segment graph** is constructed with the sparse data to observe the approximate flow of traffic between bus stops at a point in time. Each node in the graph represents a segment and holds the average speed in a 10 minute interval. It is a directed graph, and there exists an edge between two segments if they share a common bus stop

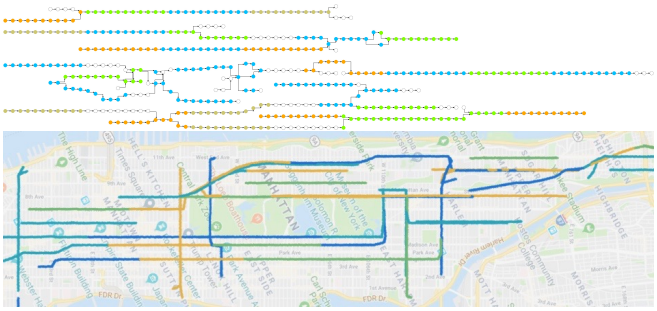

Fig. 1. Traffic graph defined over Manhattan divided into chunks of 5-hop subgraphs on which models are evaluated (each color is a subgraph). 5-hop subgraphs consist of all segments within five upstream and five downstream hops based around a selected root segment.

or, equivalently, if they are adjacent to one another. Figure 1 shows a graph of the traffic segments in Manhattan.

4. APPROACH

4.1. Mixed-Adjacency Recurrent Neural Net

We define a Mixed-Adjacency Recurrent Neural Net (*MXRNN*) to investigate the capability of Recurrent Neural Nets to disambiguate interactions between input measurements belonging to different nodes with no knowledge of their spatial connectivity. This serves as a baseline performance for us when studying neural network approaches. Measurements in each subgraph are flattened into a 1-dimensional vector by fixing the order of nodes. As the initial choice of ordering can be arbitrary, no prior adjacency information is given to MXRNN. In other words, MXRNN consists of an RNN for each subgraph, with readings from all input segments flattened out before feeding into the RNN. The spatial interactions of traffic and dependencies are learned during training. We use Long-Short Term Memory (LSTM) cells which aid in detecting long-term dependencies [14] and have been shown to work well for time series prediction models.

4.2. Message-Passing Recurrent Neural Net

We introduce greater supervision and regularization steps to improve generalization ability from the naïve MXRNN in the form of a "graphically aware" message-passing neural net called the **Message-Passing Recurrent Neural Net** (*MPRNN*). The MPRNN is a deep neural network architecture with differentiable operations which iterate message-passing among connected nodes in our traffic graph and a recurrent architecture to detect temporal effects. Through tunable model parameters, MPRNN explicitly controls the breadth of information propagation between connected nodes in the graph and the flow of information based on the direc-

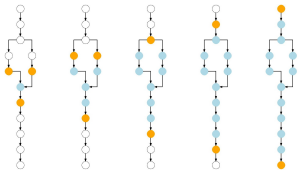

Fig. 2. *Procedural Training:* Parameters trained for smaller subgraphs are transferred to initialize training progressively larger subgraphs of $k = 1 \ldots 5$.

Model	k=1	k=2	k=3
Root-mean squared error (RMSE) (mph)			
Linear	0.505 ± 0.7	0.647 ± 0.8	0.752 ± 1.1
MXRNN	0.283 ± 0.3	0.291 ± 0.4	0.672 ± 1.8
GCN [17]	-	0.294 ± 0.3	0.294 ± 0.3
ASTGCN [6]	0.286 ± 0.3	0.274 ± 0.3	$\mathbf{0.272 \pm 0.2}$
MPRNN	$\mathbf{0.273 \pm 0.4}$	$\mathbf{0.260 \pm 0.3}$	$\mathbf{0.272 \pm 0.3}$

Table 1. Next timestep prediction ($t + 1$) accuracy evaluated on a reserved period of 18 continuous days. RMSE measurements are shown first, with Pearson Correlation Coefficient below.

tionality of node adjacencies (§3). In our results, we observe the introduction of message-passing improves generalization over baseline models (§5.2).

Departing from Graph Convolution [15], a state-maintaining messenger and LSTM unit is defined for each node $v \in V$ in the traffic graph where one node is solely responsible for learning the traffic patterns in that node given neighboring states. The set of neighbors, $N(v)$, of a node v is the set of adjacent road segments. In evaluation, each node converses in messages with their immediate neighbors based on the current observations, and then updates the internal states $\{h_t(u) \mid u \in V\}$ in its LSTM unit. At time $t + 1$, the speed is predicted from the internal state. Message-passing operates by allowing one node to observe the hidden state of its neighbors. Performing multiple iterations of message-passing allows the propagation of information beyond immediate neighbors.

4.3. Training

For both the MPRNN and MXRNN, the initial hidden state for each segment at $t = 0$, $h_0(v)$, is initialized randomly during training and evaluation, sampled from $\mathcal{N}(0, 1)$, since the previous traffic conditions are unknown. In training, loss is computed using *Mean-Squared Error (MSE)* to obtain a regression on predicted speeds. For all the LSTM cells, we picked 24 as the *history length*, or the length of unrolled LSTM network. Empirical challenges exist with back-propagating the gradient over exceedingly long histories, as gradients are distributed among more parameters [16]. Also, in the context of traffic flow, we also do not expect effects to be propagated significantly more than a few hours ahead in time. A history length of 24 corresponds to 4 hours of history (at one sample every 10 minutes), which is neither too small nor too large.

We follow a training curriculum where the model is defined and trained on smaller subgraphs then on increasingly larger graphs with transferred weights for previously trained nodes. Figure 2 shows the procedure as the number of hops-k increases. For the first 8 epochs, only the parameters defined on nodes newly introduced from $k' - 1 \rightarrow k'$ hops receives gradient update. Then, all parameters are fine tuned in a final epoch. This training procedure was determined to obtain quicker convergence for large hops and also to preserve well-performing local parameters from becoming diluted. As all segments were ensured to have $> 50\%$ data availability, a roughly equal volume of continuous data measurements of 18 days in the latter of the data collection period were reserved for testing and the remaining 72 days were used in training.

Specific to the MPRNN implementation, one iteration of message-passing dictates the propagation of one node's state to its neighbors. Multiple iterations of message-passing allows broader spread of information with tradeoffs in train-evaluation time. A fixed number of iterations occur within each time-step, thus 3 iterations were deemed appropriate in our context. As a result, newly obtained hidden states are intended to propagate only during the next timestep for segments more than 3-hops away.

5. RESULTS

5.1. Next timestep Prediction

As a first test of performance, we predict speeds in the immediate next timestep, at a distance of up to 3 hops (3 bus stops) away. This is approximately 1 kilometer. In this, we show the performance of each model in predicting the traffic speed in the test segment, 10 minutes (i.e. one timestep) ahead, given 4 hours (24 steps) of prior history. Performances are compared across a simple linear regression model with a bias term, the Mixed-Adjacency RNN (*MXRNN*), and our proposed Message-Passing RNN (*MPRNN*), and two state-of-the-art methods that employ Graph Convolution Networks, all trained and evaluated with history length 24.

As shown in Table 1, the MPRNN performs very well with a *Root Mean-Squared Error* (*RMSE*) of 0.272 mph with input from up to $k = 3$ neighboring hops, while the naïve LSTM approach, MXRNN, is comparable for low k, but then rapidly degrades in performance, indicating a lack of ability to predict at farther locations. The GCN [17] is a simple GCN that is used for supervised learning, and the ASTGCN [6] is a very recent work that beat many other state-of-the-art methods in traffic flow prediction[1].

[1] Traffic *flow* prediction is a different problem that traffic speed prediction;

Model	k=1	k=2	k=3	k=4
Root-mean squared error (RMSE) (mph)				
MXRNN	2.59 ± 1.6	2.71 ± 1.5	2.74 ± 1.8	2.85 ± 1.2
MPRNN	1.83 ± 0.3	2.13 ± 0.5	2.29 ± 0.6	2.35 ± 0.6
Pearson Correlation Coefficient (PCC)				
MXRNN	0.59 ± 0.0	0.53 ± 0.1	0.47 ± 0.1	0.47 ± 0.1
MPRNN	0.76 ± 0.0	0.64 ± 0.0	0.51 ± 0.1	0.43 ± 0.0

Table 2. Forecasting performance for increasing hops $k = 1 \dots 4$ averaged across 47 graphs which consist the traffic graph of Manhattan. Variance is reported over the individual graphs.

Model	k=1	k=3	k=5	k=7	k=9
Root-mean squared error (RMSE) (mph)					
MXRNN	3.06	3.28	4.41	5.24	4.27
MPRNN	2.75	3.26	3.25	3.27	3.24
Pearson Correlation Coefficient (PCC)					
MXRNN	0.37	0.33	0.34	0.01	0.08
MPRNN	0.50	0.30	0.19	0.16	0.13

Table 3. Forecasting error for a broader range of hops $k = 1 \dots 10$ for a the single segment plotted in Figure 1 (the intersection of *54th St and 7th Ave*)

The best-fit error rates are obtained by MPRNN, consistently demonstrating error lower than the other methods. It is noted that the MXRNN shows a sharp degradation at $k = 3$. While an outlier, the inconsistency in prediction behavior becomes much more apparent in forecasting where the MXRNN exhibits higher error rates with high variance. In general, we attribute the improvement in prediction to the regularization introduced in message-passing as spatial consistency is maintained along the true layout of traffic. This type of regularization is achieved by the graph convolutional approaches as well, which are also resilient to the addition of more hops in the input. In fact, the ASTGCN performs *better* as more hops are added. But the problem appears in forecasting, which is a much more expensive computational operation, owing to the number of parameters and variables involved. Thus the GCN based methods, which are much more complex than the other methods, take prohibitively long times to produce forecasting results, and hence we do not consider those results in our comparison.

For all implementations, we experience a tradeoff in accuracy when training for larger k-hops. At the benefit of observing information from more segments, the number of parameters increase and the training objective requires the model to fit neighboring data measurements. However, despite this, benefit of training for larger k enables forecasting over much broader segments of road.

flow refers to volume of traffic

5.2. Forecasting

Forecasting performance for subgraphs in the directed traffic graph of Manhattan are assessed given known values at entry and exit nodes. Experiments are performed for varying sizes of the subgraphs of hops $k = 1 \dots 3$ where the tested models must propagate known observations over an increasing sequence of intermediate segments for which observations are not available. All forecasting errors were assessed for the reserved period of 18 days. Table 2 shows the forecasting errors (RMSE) across all subgraphs in Manhattan for $k = 1 \dots 3$. In addition to the RMSE metric, the Pearson Correlation Coefficient (*PCC*) [18], which characterizes the quantifies the correlation between the predicted and real values, is shown. For one specific segment, we also show the errors for larger values of k up to 9 in table 3. For the first available time step of each day, both MXRNNs and MPRNNs were initialized with zeros (mean of random initialization during training). Forecasting then proceeded for all subsequent traffic speeds throughout the day until the last available measurement.

As can be seen from the tables, the MPRNN outperforms the MXRNN in all cases, and particularly when using larger number of hops. While the ability to forecast future values degrade for all models, there is a more gradual degradation in error for the MPRNN (Table 3), where for hops $k = 9$, the MPRNN still maintains an RMSE of 3.24 mph while using the MXRNN produces an error of 4.27 mph with indications that error will continue to increase with more hops. We reiterate that the MPRNN architecture only defines message propagation which is consistent with the spatial layout of the traffic graph, as opposed to MXRNN which is not regularized by spatial information and thus defines arbitrary correlations with fully-connected layers over any input measurements.

6. CONCLUSION

Highly parameterized neural nets have been applied successfully to data that is sporadic and unidimensional, but abundant and easily collected at the same time. We examine historical travel data of public transit vehicles in New York City which currently sees use solely to check bus arrival times for commuters. We aggregate bus speeds into a traffic segment graph that represents the relationships between road segments. The MPRNN architecture is defined on the graphical representation of traffic flow which leverages the interaction between pairs of traffic segments where flow patterns of individual segments are subject to highly variable factors. Forecasting performance is assessed on traffic segments spanning the entirety of Manhattan. Benchmarks are presented in fine tuning and in close comparison with a more naïve implementation of MPRNN, called MXRNN, resulting in a final model which produces meaningful forecasts several hours ahead in time.

7. REFERENCES

[1] Dave Barth, "The bright side of sitting in traffic: Crowdsourcing road congestion data," Aug 2009.

[2] Resty Woro Yuniar, "Google maps: a lost cause for indonesian drivers," Apr 2018.

[3] Zhongjian Lv, Jiajie Xu, Kai Zheng, Hongzhi Yin, Pengpeng Zhao, and Xiaofang Zhou, "LC-RNN: A deep learning model for traffic speed prediction," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*. 7 2018, pp. 3470–3476, International Joint Conferences on Artificial Intelligence Organization.

[4] Zhiyong Cui, Kristian Henrickson, Ruimin Ke, and Yinhai Wang, "High-order graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting," *arXiv preprint arXiv:1802.07007*, 2018.

[5] Avinash Achar, Venkatesh Sarangan, Rohith Regikumar, and Anand Sivasubramaniam, "Predicting vehicular travel times by modeling heterogeneous influences between arterial roads," in *The 32nd AAAI Conference on Artificial Intelligence*, 2018.

[6] Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in *AAAI*, 2019.

[7] Bing Yu, Haoteng Yin, and Zhanxing Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*. July 2018, International Joint Conferences on Artificial Intelligence Organization.

[8] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *International Conference on Learning Representations*, 2018.

[9] Eleni I. Vlahogianni, Matthew G. Karlaftis, and John C. Golias, "Short-term traffic forecasting: Where we are and where we're going," *Transportation Research Part C: Emerging Technologies*, vol. 43, pp. 3–19, jun 2014.

[10] Toru Seo, Alexandre M. Bayen, Takahiko Kusakabe, and Yasuo Asakura, "Traffic state estimation on highway: A comprehensive survey," *Annual Reviews in Control*, vol. 43, pp. 128–151, 2017.

[11] M. Lippi, M. Bertini, and P. Frasconi, "Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 2, pp. 871–882, June 2013.

[12] Wei Liu, Yu Zheng, Sanjay Chawla, Jing Yuan, and Xing Xie, "Discovering spatio-temporal causal interactions in traffic data streams," in *17th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2011)*, August 2011.

[13] NYC MTA, "MTA Bus Time Historical Data," http://web.mta.info/developers/MTA-Bus-Time-historical-data.html, 2014.

[14] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[15] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl, "Neural message passing for quantum chemistry," in *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017, vol. 70, pp. 1263–1272.

[16] S. Hochreiter, "Untersuchungen zu dynamischen neuronalen Netzen. Diploma thesis, Institut für Informatik, Lehrstuhl Prof. Brauer, Technische Universität München," 1991.

[17] Thomas N Kipf and Max Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[18] Joseph Lee Rodgers and W. Alan Nicewander, "Thirteen ways to look at the correlation coefficient," *The American Statistician*, vol. 42, no. 1, pp. 59–66, 1988.