Estimating Traffic Density Using Transformer Decoders

Wang, Yinsong; Zhang, Jing; Nikovski, Daniel; Kojima, Takuro

TR2023-011 March 17, 2023

Abstract

We propose a combined particle-based density prediction model consisting of three components: trajectory prediction for existing particles, entering particle prediction, and iterative sampling. At initialization, the combined model takes in a set of trajectories for trajectory prediction and a sequence of observation vectors for entering particle prediction. Then, the iterative sampling module generates the density prediction for the next time instance. It will also sample a pool of particles and pass on their trajectories to the next trajectory prediction model for future density prediction.

International Workshop on Statistical Methods and Artificial Intelligence 2023

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2023) 000–000

Procedia
Computer Science

www.elsevier.com/locate/procedia

The 4th International Workshop on Statistical Methods and Artificial Intelligence (IWSMAI'23)
March 15-17, 2023, Leuven, Belgium

Estimating Traffic Density Using Transformer Decoders

Yinsong Wang[1 a,*], Jing Zhang[b], Daniel Nikovski[b], Takuro Kojima[c]

[a]*Northeastern University*
wang.yinso@northeastern.edu
[b]*Mitsubishi Electric Research Labs*
{jingzhang, nikovski}@merl.com
[c]*Mitsubishi Electric Corporation*
kojima.takuro@ah.mitsubishielectric.co.jp

Abstract

We propose a combined particle-based density prediction model consisting of three components: trajectory prediction for existing particles, entering particle prediction, and iterative sampling. At initialization, the combined model takes in a set of trajectories for trajectory prediction and a sequence of observation vectors for entering particle prediction. Then, the iterative sampling module generates the density prediction for the next time instance. It will also sample a pool of particles and pass on their trajectories to the next trajectory prediction model for future density prediction.

Keywords: traffic density estimation; transformer; transportation network; deep learning

1. Introduction

Modeling and prediction of pedestrian and vehicular traffic has many importatnt applications, such as optimal vehicle guidance in transportation networks, optimal group elevator scheduling in buildings, crowd management in airports, malls, and train stations, prevention of disease transmission in public areas, etc. Such traffic models are usually represented by sets of origin, destination, and intermediate locations, such that traffic participants (vehicles, pedestrians, etc.) originate in one of the origin locations and travel to their intended destination location through a series of intermediate locations. A traditional modeling tool is the origin/destination (OD) matrix of the transportation

[1]This work was done during an internship at Mitsubishi Electric Research Labs (MERL).
*Corresponding author. Tel.: +1-979-739-9710
E-mail address: wang.yinso@northeastern.edu

network whose entries express the arrival rate for each pair of origin/destination nodes in the network. This matrix can be estimated from measured data, as long as the origin and destination of each trip in the network can be recorded.

The OD matrix can be very useful for many purposes, such as road planning and determining the number of elevators in a building. However, it does not provide enough information about where traffic participants can be at any moment in time, which is necessary if overcrowding, congestion, and possible contagion in particular areas is to be mitigated. For these purposes, some measure of the traffic density needs to be computed. Given the stochastic nature of traffic, it is clear that such traffic density should be probabilistic in nature.

In road transportation networks, this problem is usually solved with the help of digital roadmaps that contain the transportation network graph. Macro simulators simulate traffic flow directly, whereas micro simulators simulate the movement of many vehicles, sampled from the OD matrix. Both kinds of simulators make some assumptions about what kind of path a vehicle would take between a pair of origin/destination nodes, as well as how drivers would react to congestion. The accuracy of such simulators depends critically on how accurate these assumptions are.

In contrast, usually there are no digital roadmaps for buildings and other indoor spaces, where pedestrians are free to move around between any unobstructed spaces. In such cases, predicting where pedestrians are going to be is a very challenging problem. In this paper, we address it by learning a predictive model of passenger movement, and use it for particle-based simulation of traffic, similar to micro simulators of vehicular traffic. In learning the predictive model, we make the additional assumption that not only the origin and destination nodes of every trip are known, but also the entire trajectory connecting them. This kind of assumption is increasingly justified in transportation scenarios, and has been enabled technically by tracking technology such as GPS for vehicles and indoor tracking in buildings using WiFi signals, employee ID cards, vision, and other sensor technologies [1].

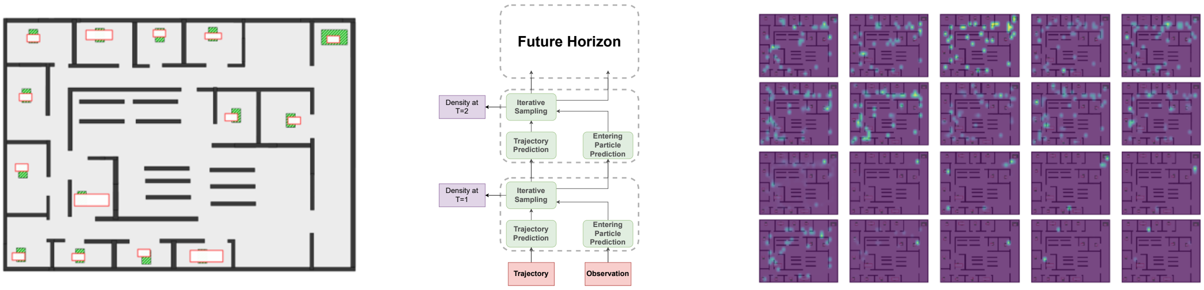
Given such extra trajectory data, it is no longer necessary to make assumptions of questionable validity and accuracy about what path a traffic participant will take between a pair of origin/destination nodes — instead, the pattern of movement and its variability can be learned from data, if the right machine learning (ML) tools are used. What ML tools to use is by no means a trivial question. We want to predict future movement from past trajectories, and there are several complications due to the nature of traffic. First, traffic is stochastic — for a given past trajectory, there are usually multiple possible continuations of this trajectory, corresponding to the different destinations this trip might have, as well as the different paths available to get there. So, predictions must necessarily be probabilistic in nature, which disqualifies many ML algorithms that produce only point estimates of their output variables. Second, movement is not first-order Markovian, as the current location of the traveller, which is the last location in the sequence of past states, does not contain all available information for predicting the future movement. If the traveller is headed in a particular direction, as a result of trying to get to a specific chosen destination, this information is encoded in the last two positions, as their difference is a measure of that direction. Earlier locations in the sequence might be informative, too, as they might indicate who the traveller is, which would be highly useful for the purposes of prediction, if this traveler typically goes to only few destinations.

This kind of issues has been addressed in the research literature on the related problem of destination prediction. The destination prediction problem has been solved mostly by either model-based [4] or deep learning-based [5] approaches. Hidden Markov Models (HMM) were applied in [7] to predict driver intent and destination. Deep learning-based methods have become popular recently because of their capability of working with long sequence inputs. For example, Recurrent Neural Networks (RNN) were used for destination prediction in [6]. To deal with gradient vanishing/exploding issues [11] of RNNs, Long-Short Term Memory (LSTM) [8] was designed to allow for long term dependency retention. In [15], LSTM models were used for trajectory prediction tasks.

In this work, we consider the traffic density estimation problem. Our contributions are as follows: 1) We propose a combined particle-based traffic density prediction model consisting of three components: trajectory prediction for existing particles, entering particle prediction, and iterative sampling. 2) We show that this model allows for an iterative prediction process that can predict over infinite horizons. 3) We conduct an extensive empirical evaluation using simulated traffic data, which validates the effectiveness and efficiency of our approach.

2. Related Work

Traffic Density Prediction: Traffic density prediction is an important subject for congestion prediction and route planning. Traffic prediction methods can be classified into model-driven and data-driven [2]. Model-driven methods



(a) (b) (c)

Fig. 1. (a) A system of interest; (b) Combined model; (c) Model prediction visualization for 5 consecutive moments in time, each 10 time units apart. First Row: Ground truth density map. Second Row: Proposed model (this work) prediction. Third Row: Transformer decoder baseline prediction. Fourth Row: GCRN prediction.

rely heavily on a network topology that can correctly describe the system of interest. For example, DynaMIT [3], Dynemo [13], VISTA [19] are all simulation models that utilize a pre-defined network topology and system logic to simulate the traffic condition. This line of work, although accurate when the underlying system logic and parameters are properly modeled, requires a laborious tuning and calibration process and significant human involvement to produce a good network and system logic. With the development of sensor technology, data-driven traffic density prediction has become more and more popular for short-term traffic density prediction. In [10], nearest neighbor regression in the temporal domain is used to predict particle locations, and in turn, predict the traffic density over a map. In [9], possible particle trajectories are modeled as a collection of shortest paths that connect the particles to their possible destinations. Similar to model-driven methods, this method also relies heavily on a pre-defined network with fixed edges representing the possible route. The conditional probability, on the other hand, is learned in a data-driven way. The method in [18] directly works on the traffic speed data, which is transformed into crowdedness data (low speed indicates high crowdedness) and models the crowdedness progression with a capped time successive diffusion model.

Deep Sequential Prediction: Deep time series prediction has two most popular subclasses: transformer models [17] and recurrent networks, including long-short term memory models [8]. The Transformer Network was first introduced in [17] as a sequence transduction system that has been shown to outperform preceding state-of-the-art methods (RNN, LSTM, etc.) in natural language processing [12]. This system is devoid of any recurrent cell units present in RNN models and instead relies on a more effective *attention* module for relating elements in a sequence. In [16], a Transformer was used to predict destinations in a contextless data setting where solely positioning information is used to make informed destination predictions.

3. Preliminaries

Throughout this paper, we use regular (non-bold) letters to represent scalars, e.g. x, bold lower-case letters to represent vectors, e.g. $\mathbf{x}$, and bold capital letters to represent matrices, e.g. $\mathbf{X}$. We use $[1, \ldots, m]$ to represent a set of integers from 1 to m. We denote by $|\mathbf{x}|$ the ℓ_1 norm of a vector $\mathbf{x}$, and $\mathbf{x}[j]$ the jth element in a vector $\mathbf{x}$.

We consider a system with confined space and predefined entry/exit locations. Particles are allowed to enter or leave the system at any time. Figure 1a is an example of our system of interest. We divide the system space into discrete grid points to describe the locations of particles. The grids (locations) are represented with $x \in [1, \ldots, m]$, where m is the total number of grid cells. Given our space partition, we describe the density map over the system with a multinomial density vector $\mathbf{x}^{DM} \in \mathbb{R}^m$ with ℓ_1 norm $|\mathbf{x}^{DM}| = 1$. We also discretize the time representation to describe a trajectory with a sequence of location indices $\{x_i \in [1, \ldots, m]\}_{i=1}^{T}$, where T is the length of the trajectory.

4. Model and Algorithm

In this section, we propose the Transformer Decoder-based traffic density prediction model. The model has three major components: trajectory prediction for existing particles, entering particle prediction, and iterative sampling. The three components allow for traffic density prediction over future horizons.

4.1. Trajectory Prediction for Existing Particles

The original Transformer [17] requires an encoder to encode linguistic information in the source language and translate it into a new language with a decoder. In our application, we formulate the trajectory prediction problem as a sentence completion problem, which is equivalent to a translation problem without source language encoding.

The location index sequence $\{x_i \in [1, \dots, m]\}_{i=1}^{T}$ will first go through embedding and positional encoding layer to obtain its high dimensional representation. The representation will then go through several Transformer decoder layers to accumulate temporal information. The first Attention module will be masked to ensure the model only makes predictions based on seen partial trajectories. The target outputs are the subsequent location indices $\{x_i \in [1, \dots, m]\}_{i=2}^{T+1}$. The model is trained with cross-entropy loss and the model output can represent the multinomial location distribution after a softmax function, which we denote by $\hat{\mathbf{x}}$. Similar to language models, we also designate location index 0 as an exit indicator.

4.2. Entering Particle Prediction

The trajectory prediction model can account for the density map of all existing particles in the system. However, for a system with constant entering and exiting particles, the model should also account for entering particle predictions to make an accurate density map prediction into the future horizon. We propose a further skimmed version of the Transformer decoder model for this task.

The model takes a sequence of observation vectors $\{\mathbf{z}_i \in [0, 1]^m\}_{i=1}^{T}$ as input and feeds it directly to the transformer decoder module for temporal information accumulation. Each element of the observation vector is a binary variable with 1 indicating there is an entering particle and 0 for no observation. Again, the first attention module is masked to prevent temporal information back-tracking. The target output are the subsequent observation vectors $\{\mathbf{z}_i \in [0, 1]^m\}_{i=2}^{T+1}$. The last layer of the observation prediction model is a sigmoid layer to make each element in the output $\hat{\mathbf{z}} \in \mathbb{R}^m$ a proper probability indicating the possibility of observing an entering particle at this location. The model is trained with mean-squared error loss.

4.3. Iterative Sampling

Iterative prediction needs not only two prediction models that account for particles in the system and particles entering the system but also an iterative sampling mechanism. We propose a simple iterative sampling module that allows density predictions into infinite future horizons.

At any point in time, there will be a set of n existing particles in the system. The trajectory prediction model will use the set of trajectories associated with these particles to output a set of multinomial vectors $\{\hat{\mathbf{x}}_i \in \mathbb{R}^m\}_{i=1}^{n}$ for the next possible locations. The entering particle prediction model will also yield a probability vector $\hat{\mathbf{z}}$. The average module will first return the density prediction $\mathbf{p}$ at this time as $\mathbf{p}[j] = \frac{\sum_{i=1}^{n} \hat{\mathbf{x}}_i[j] + \hat{\mathbf{z}}[j]}{|\sum_{i=1}^{n} \hat{\mathbf{x}}_i + \hat{\mathbf{z}}|}, \quad j \in [1, \dots, m]$, which is an averaged vector of the trajectory predictions and entering particle predictions.

After density prediction, the average module will pass on the trajectory predictions to a sampling module to create a pool of possible particle trajectories for existing particles. For example, if we want to initiate the future horizon prediction with a pool of particle trajectories, each trajectory will be created as follows: i) Sample one particle from the existing n particles with equal probability. ii) For the sampled particle, sample a next location index based on the output multinomial vector $\hat{\mathbf{x}}$ of this particle. iii) Append the location index to the original trajectory $\{x_i \in [1, \dots, m]\}_{i=1}^{T}$ to create a new trajectory $\{x_i \in [1, \dots, m]\}_{i=1}^{T+1}$. On the other hand, we use another sampling module to create a set of entering particles according to output probability vector from the entering particle prediction model. Each sample creates a new trajectory of length one (the entry location). The two sets of trajectories will be combined and fed to the next trajectory prediction model for future prediction. The new set of entering particles will also create a new observation vector that will pass on to the next entering particle prediction model for future prediction.

Combining all three components together, we finally propose our density prediction model as a whole. The model is shown in Figure 1b. At initialization, the combined model takes in a set of trajectories for trajectory prediction and a sequence of observation vectors for entering particle prediction. Then, the iterative sampling module generates the density prediction for time $T = 1$. It will also sample a pool of particles and pass on their trajectories to the next trajectory prediction model for future density prediction. This model allows for an iterative prediction process that can predict over infinite horizons.

5. Empirical Evaluation

In this section, we examine the advantages of incorporating trajectory information into density prediction by comparing our proposed model to other well-established sequential prediction models. We create a large synthetic simulation data set for this comparison using our example system shown in Figure 1a.

5.1. Benchmark Algorithms

We compare our model with three benchmark algorithms, including a transformer baseline model and a state-of-the-art time series prediction model. **Transformer Decoder:** The baseline Transformer Decoder model is structurally identical to the entering particle model. The model takes a sequence of multinomial density vectors $\{\mathbf{p}_i\}_{i=1}^{T}$ as input, and target subsequent multinomial density vectors $\{\mathbf{p}_i\}_{i=2}^{T+1}$ as output. **Graph Convolutional Recurrent Networks (GCRN) [14]:** GCRN captures spatio-temporal information with its graph convolution and recurrent neural network respectively. It also takes multinomial density vectors $\{\mathbf{p}_i\}_{i=1}^{T}$ as its input node features, and targets subsequent vectors as output. Compared to the Transformer Decoder baseline model, GCRN captures the structural information with a graph representation.

5.2. Data Set

We have created a simulation data set for the empirical evaluation using Vectorworks software SimTread module. The simulation is carried out in our example system in Section 3. The probabilistic linkage between the entry locations and exit locations is uniquely defined by a randomly generated 14 by 14 origin-destination matrix. The ground truth particle generation process, which individually is a Poisson process at each location, is defined according to the size of the entry blocks in the system diagram (see Fig. 1a). This design is similar to a realistic floor plan in a typical office building. Our synthetic data set (referred to as the SimTread data set) contains two sets of data, training and testing. The training set has 2036 particles and a time span of 5110. The particle location recording interval is 0.1 seconds, equal to a total of 511 seconds of data. Time 5110 marks the last particle leaving the system. The testing set follows an identical origin-destination matrix and particle generation rate as the training set. It also contains 2036 particles, but with a time span of 4986. It has the same recording interval as the training set, equal to 498.6 seconds of data.

5.3. Performance

The prediction accuracy comparison is shown in Table 1. We compare the prediction accuracy over a horizon of length of 50 time steps. All models show an increase in the prediction error as the prediction horizon increases. The transformer decoder baseline model shows the worst performance among all models. Compared to the state-of-the-art GCRN model, our proposed model shows a consistent performance advantage between 20% to 30%. Our proposed model also shows a slow decay in prediction accuracy over the prediction horizon as well.

5.4. Visualization

In addition to the performance advantage in terms of prediction accuracy, we also visualize the prediction difference between the benchmark models and our proposed model to demonstrate the fundamental differences in the predictions

Model	Time 1	Time 10	Time 20	Time 30	Time 40	Time 50
Transformer Decoder	8.52×10^{-4}	9.66×10^{-4}	10.07×10^{-4}	10.25×10^{-4}	10.29×10^{-4}	10.15×10^{-4}
GCRN	1.55×10^{-5}	7.41×10^{-5}	7.50×10^{-5}	7.97×10^{-5}	8.24×10^{-5}	8.24×10^{-5}
Proposed Model	$\mathbf{1.11 \times 10^{-5}}$	$\mathbf{3.54 \times 10^{-5}}$	$\mathbf{4.78 \times 10^{-5}}$	$\mathbf{5.83 \times 10^{-5}}$	$\mathbf{5.77 \times 10^{-5}}$	$\mathbf{6.45 \times 10^{-5}}$

Table 1. Prediction Mean Squared Error over 50 prediction horizon (lower is better).

further. In Figure 1c, we show the predicted density map for the Transformer Decoder baseline model, GCRN model, and our proposed model. The last row is the corresponding ground true density map. The density map is interpolated bilinearly from 30×20 to 1500×1200 for better visual resolution and quality. Figure 1c shows that in addition to the statistical advantage, our proposed model also provides a prediction structure closer to reality.

6. Conclusion

In this work, we have proposed a combined traffic density prediction model that predicts traffic density maps from an individual trajectory level. Compared to traditional approaches, our model can take advantage of particle trajectories collected from sensor readings. Simulations have shown that our proposed model can outperform state-of-the-art map-based density prediction algorithms.

References

[1] Adib, F., Kabelac, Z., Katabi, D., Miller, R.C., 2014. 3D tracking via body radio reflections, in: Proceedings of the 11th USENIX Conference on Networked Systems Design and Implementation, pp. 317–329. URL: `https://dl.acm.org/doi/abs/10.5555/2616448.2616478`.

[2] Barros, J., Araujo, M., Rossetti, R.J., 2015. Short-term real-time traffic prediction methods: A survey, in: 2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS), IEEE. pp. 132–139.

[3] Ben-Akiva, M., Bierlaire, M., Koutsopoulos, H., Mishalani, R., 1998. Dynamit: a simulation-based system for traffic prediction, in: DACCORD short term forecasting workshop, Delft The Netherlands.

[4] Besse, P.C., Guillouet, B., Loubes, J., Royer, F., 2018. Destination prediction by trajectory distribution-based model, in: Proceedings of the 21st IEEE International Conference on Intelligent Transportation Systems.

[5] Choi, S., Kim, J., Yeo, H., 2019. Attention-based recurrent neural network for urban vehicle trajectory prediction. Procedia Computer Science 151, 327–334.

[6] Endo, Y., Nishida, K., Toda, H., Sawada, H., 2017. Predicting destinations from partial trajectories using recurrent neural network, in: Proceedings of the 21st Pacific-Asia Conference on Knowledge Discovery and Data Mining.

[7] Epperlein, J.P., Monteil, J., Liu, M., Gu, Y., Zhuk, S., Shorten, R., 2018. Bayesian classifier for route prediction with markov chains, in: Proceedings of the 21st IEEE International Conference on Intelligent Transportation Systems.

[8] Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Computation 9, 1735–1780.

[9] Kriegel, H.P., Renz, M., Schubert, M., Zuefle, A., 2008. Statistical density prediction in traffic networks, in: Proceedings of the 2008 SIAM International Conference on Data Mining, SIAM. pp. 692–703.

[10] Oswald, R.K., Scherer, W.T., Smith, B.L., 2000. Traffic flow forecasting using approximate nearest neighbor nonparametric regression. Technical Report.

[11] Pascanu, R., Mikolov, T., Bengio, Y., 2013. On the difficulty of training recurrent neural networks, in: Proceedings of the 30th International Conference on Machine Learning (ICML).

[12] Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., 2018. Improving language understanding by generative pre-training, in: OpenAI.

[13] Schwerdtfeger, T., 1984. Dynemo: A model for the simulation of traffic flow in motorway networks, in: Proceedings of the 9th international Symposium on Transportation and Traffic Theory, VNU Science Press Utrecht, The Netherlands. pp. 65–87.

[14] Seo, Y., Defferrard, M., Vandergheynst, P., Bresson, X., 2018. Structured sequence modeling with graph convolutional recurrent networks, in: International conference on neural information processing, Springer. pp. 362–373.

[15] Shi, X., Shao, X., Guo, Z., Wu, G., Zhang, H., Shibasaki, R., 2019. Pedestrian trajectory prediction in extremely crowded scenarios. Sensors .

[16] Tsiligkaridis, A., Zhang, J., Taguchi, H., Nikovski, D., 2020. Personalized destination prediction using transformers in a contextless data setting, in: Proceedings of the 2020 International Joint Conference on Neural Networks.

[17] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I., 2017. Attention is all you need, in: Proceedings of the 31st Conference on Neural Information Processing Systems.

[18] Zhao, B., Xu, C., Liu, S., 2017. A data-driven congestion diffusion model for characterizing traffic in metrocity scales, in: 2017 IEEE International Conference on Big Data (Big Data), IEEE. pp. 1243–1252.

[19] Ziliaskopoulos, A.K., Waller, S.T., 2000. An internet-based geographic information system that integrates data, models and users for transportation applications. Transportation Research Part C: Emerging Technologies 8, 427–444.