Transfer Learning for Traffic Speed Prediction: A Preliminary Study

Bill Y. Lin,[*] **Frank F. Xu,**[*] **Eve Q. Liao, Kenny Q. Zhu**

{yuchenlin, frankxu, liaoqz}@sjtu.edu.cn, kzhu@cs.sjtu.edu.cn
Department of Computer Science and Engineering
Shanghai Jiao Tong University, Shanghai, China

Abstract

Traffic speed prediction can benefit a wide range of IoT applications in intelligent transportation and smart city. Recent supervised machine learning approaches heavily leverage vast amount of historical time-series data. Consequently, they degrade dramatically in the areas where collecting a large traffic data is not quite feasible. With the aim of predicting the traffic speed of such urban areas, we propose a transfer learning framework that exploits historical data of some other data-abundant areas by utilizing various spatio-temporal semantic features. Experimental results show that classic regression models and our proposed kernel regression model can achieve competitive performance comparing to baseline methods that heavily rely on the historical data of target areas.

Introduction

Traffic speed prediction is a challenging problem and has various downstream applications of Internet of Things (IoT), many of which are fundamental to intelligent transportation systems and smart city, such as congestion management, urban planning, vehicle routing, etc. (Pan, Demiryurek, and Shahabi 2012; Xu et al. 2015; Mchugh 2015).

Most existing machine learning approaches heavily rely on the vast amount of historical data for the areas being predicted (Ren et al. 2014; Clark 2003). However, accurate and reliable historical traffic data collected from road sensors is very expensive and available in urban areas where the government can afford the large cost. Consequently, most state-of-the-art time-series based models cannot be applied directly on areas where little traffic data is available.

Another disadvantage of most existing approaches is that they only focus on temporal features and do not explicitly utilize semantic features from spatio-temporal patterns (Yao et al. 2017; Lin et al. 2017), which benefit many practical applications of urban computing (Zheng et al. 2014). Research on extracting such effective spatial features for traffic speed prediction is almost missing from the literature.

The preliminary study aims to answer this research questions: *How can we exploit the data of data-abundant areas to predict traffic speed for areas without traffic data through their semantic spatio-temporal features?* To the best of our

knowledge, we are among the first to study transfer learning for traffic speed prediction (Xu et al. 2016). The contributions of this paper are as follows:

- We extract various spatial features in multiple levels and combine them with temporal features to support this transfer learning scenario, which also improves the transparency of prediction models.

- Based on the features, we propose a novel clustering-based transfer learning model. Experimental results show that proposed model perform competitively with classic regression methods, but using only distant supervision.

Transfer Learning Scenario

Consider we have a set of n road segments with traffic speed sensors. At any given time t, each sensor i provides a traffic speed reading at this current time, denoted as $v_i[t]$. Traffic speed prediction problem is to predict a future traffic speed like $v_i[t + h]$ at a previous time t. Most models utilize historical traffic data to predict the future speed of the same areas. However, when historical data such as $v_i[1 : t]$ is not available, it is infeasible for them to predict.

In this scenario, a transfer learning (Pan and Yang 2010) approach is supposed to exploit the data of some *source areas* $\mathcal{S}$ to build a prediction model for other *target* areas $\mathcal{T}$, where there is little traffic speed data. Mathematically, given $\{v_i[1 : t] | i \in \mathcal{S}\}$, a transfer learning model is expected to be able to predict $\{v_j[1 : t] | j \in \mathcal{T}\}$.

Traffic Estimates Dataset

In this section, we briefly introduce a public dataset named *UIUC New York City Traffic Estimates*[1], on which we extract spatial features and conduct our following experiments. This dataset covers 700 million trips from 2010 to 2013 in New York City. It contains accurate hourly traffic speed measurement for almost all individual links of the NYC road networks. Specific data format is described as follows:

1. the road network is represented as a directed graph composed of *nodes* and *links*;

2. each *node* is an intersection of the road network, with multiple properties like latitude and longitude;

---

[*]The first two authors contribute equally.

[1]https://publish.illinois.edu/dbwork/open-data/

	Hudson	Manhattan	Brooklyn	Bronx	Queens
#link	730	8,578	7,790	2,113	8,173
#trips_sum	477k	52m	24m	5m	20m

3. each directed *link* is a small road segment connecting two such nodes;

4. generally, a real street consists of multiple links; two-way streets are often represented as two directed links with opposite directions;

5. each row of the traffic speed data is the average traffic speed of a particular link at a particular hour.

To evaluate transfer learning approaches, we split the road network into five different areas as shown in Table 1.

Spatiotemporal Features

In this section, we discuss the proposed spatial features extracted from OpenStreetMap[2] and temporal features that act as fundamental components of our proposed transfer learning approach. The proposed spatial features capture the traffic-related geographical characteristics for each link in road networks.

Basic Information Features

An example of extracting the basic information for a particular link is shown in Figure 1. We have 5 features for representing the basic information of each link: `length`, `#begin_node_in_links`, `#begin_node_out_links`, `#end_node_in_links` and `#end_node_out_links`. For each link, the `length` is the real distance between the begin and the end node of this link. The other 4 features represent the number of *in* and *out* links connected to both nodes of a link, which may provide information about crossroads or one-ways.

Figure 1: An example for extracting basic information features.

Road Density Features

Additionally, we believe traffic speed is highly relevant to road density, which can be measured by the number
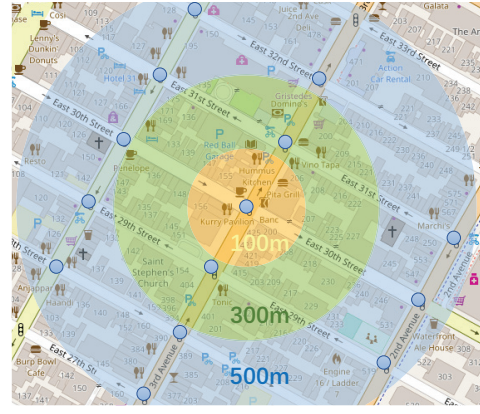
[2]https://www.openstreetmap.com

Figure 2: An example of extracting the road density features with three different radius. The origin is begin/end node of a certain link, and each blue circle is a link intersection.

of neighboring nodes and links within the same area. To be more specific and capture the sensitivity about directions, we compute road density respectively for each end in terms of the density of neighboring node , and the density of neighboring in and out link, according to three radius (100/300/500m), as shown in Figure 2. Consequently, we have $2 \times 3 \times 3 = 18$ road density features in total.

Categorical POI Density Features

Points of interest (POI) are specific locations that people may find useful or interesting, such as restaurants, shopping halls, parks, etc. Since such places are very influential to the traffic, we query nearby POIs[3] for each node with three different radius (100/300/500m) using HERE Places API[4]. Figure 3 shows such an example for extract *road density* features and *POI density features*.

Temporal Features

Our temporal feature is simply a distributed representation of the time information. It is basically a concatenation of several one-hot vectors, where each vector represents the *month*, the *day* of a week, the *hour* of a day and whether it is *workday* respectively.

Transfer Learning Approach

Obtaining the above features for link and time, we first apply several classic machine learning models for regression (Linear Regression, Neural Network model[5], Support Vector Regression). They are trained on source areas with above-explored spatio-temporal features and then predict traffic condition on target areas as a test. Afterwards, we present a novel transfer learning approach named CTMP.

[3]The 11 major POI types we consider are {*Eat, Drink, Going Out, Sights & Museums, Transport, Accommodation, Shopping, Business & Services, Facilities, Facility, Administrative Areas/Buildings, Natural or Geographical*}.

[4]https://developer.here.com/documentation/places/topics/introduction.html

[5]A two-layer fully connected network model.

Figure 3: An example for extracting POI Features. Different colors indicate different POI types.

We introduce our novel **C**lustering-based **T**ransfer **M**odel for **P**rediction (CTMP), which first clusters links in both source and target areas based on their spatial features and then do time series based prediction for the target links based on neighboring source links with historical data.

Intuition Behind the CTMP

Our intuition behind CTMP is that given a link in target areas with spatial features, we can first find the most similar links in source areas and then leverage the source data to predict the speed of links in target areas.

The assumption here is that links with similar spatial features should also share similar traffic patterns. However, simply clustering road links based on spatial features performs not very well in practice, because not all the features are equally important and the importances cannot be obtained in such an unsupervised way. Therefore, we incorporate a regularization term in the distance metric for feature reduction and selection.[6]

Clustering with Regularized Distance Metric

We use the s_i and s_j to denote two spatial feature vectors of any two links i and j respectively. We capture the distance between the two feature vectors by computing $\text{s\_dis}(i, j) = 1 - \cos(s_i, s_j)$. To regularize the time series similarities between two links, we add a regularization term $\text{t\_dis}(i, j)$, which has multiple options. A desirable option is the Dynamic Time Wrapping (DTW) (Keogh and Pazzani 2001) similarities between the weekly HAM traffic speed series of the two links. Thus, the total distance between two links can be regarded as follows, where λ is a hyper parameter to control the weight of temporal distance: $\text{dis}(i, j) = \text{s\_dis}(i, j) + \lambda \text{t\_dis}(i, j)$.

With such supervision in the source area data, we can use K-means as our clustering algorithm. For each query instance (l, t)[7], we first find the closest k neighboring

---

[6]CTMP model can be seen as a combination of clustering and Nadaraya-Watson kernel regression.

[7]The link l has no historical data in the transfer scenario.

source links with historical data $\{l_1, ... l_k\}$. We compute all the distances between them and the target link l respectively, and obtain the set of spatial feature distances $\{\text{dis}(l, l_1), ..., \text{dis}(l, l_k)\}$. Also, we can get the predicted typical traffic speed for such neighboring links based on existing time-series models at the time t: $\{y(l_1, t), ..., y(l_k, t)\}$. Finally, we can compute the predicted result for the query instance (l, t) is:

$$y(l, t) = \sum_{i=1}^{k} \left(\frac{\text{dis}(l, l_i)}{\sum_{j=1}^{k} \text{dis}(l, l_j)} y(l_i, t) \right)$$

Evaluation

To evaluate the performance of our extracted features and proposed feature-based transfer regression models, we conduct a series of experiments to check both the performance of *local transfer* and *cross-region transfer*. In this section, we first discuss the setup of our experiments, then the baseline methods and finally present the discussion of the experimental results.

Experiment Setup

We first split the data into training set and test set with respect to the time. Specifically, we first split the data in 2013 into two parts by timespan: Jan. - Jun. and Jul. - Dec. Three scenarios are shown as follows:

1. *No Transfer* task is to predict the future speed (the second half year) of a link with the historical (the first half year) data of the link.

2. *Local Transfer* task is to consider the data in first half year as training data and second half year data as test data. We train the models for each region with their data and test the models with the testing data in the same region.

3. *Cross-region Transfer* task is to use a model trained on the first half year data of a source region to predict the traffic speed of another target region in the second half year.

Baseline Methods

We compare with the most representative time-series based regression models: ARIMA (Box 2013) and HAM (Pan, Demiryurek, and Shahabi 2012).

Experimental Results

We first show the performance of HAM and ARIMA models on the *No Transfer* Task with two metrics (RMSE and MAE) in Table 2. We found that HAM performs substantially better than the ARIMA model with both metrics, which is probably due to the time interval in the dataset is one hour, quite longer than common time interval length for ARIMA model.

Also, we present the results of both *Local Transfer* (LT) and *Cross-region Transfer* (CT) in Figure 4. It can be concluded from each sub-figure that our methods achieve the similar RMSE with the HAM without explicitly using historical data for links in target areas. Also, we can find that when the two regions are similar to each other, then when we cross-region transfer one to another the performance is still
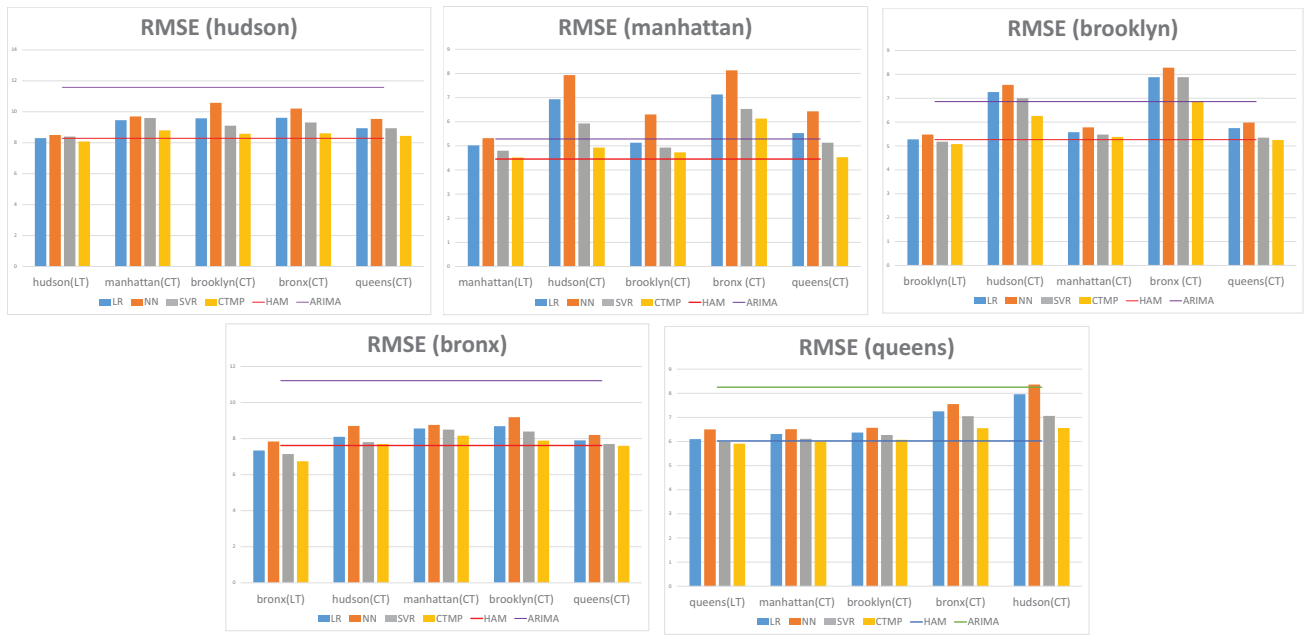
Figure 4: *Local Transfer* and *Cross-region Transfer* performances in terms of RMSE; Each sub-figure is a about region, and each cluster is either a *Local Transfer* (LT) or a *Cross-region Transfer* (CT).

Table 2: *No Transfer* performance of HAM and ARIMA

	HAM *RMSE*	HAM *MAE*	ARIMA *RMSE*	ARIMA *MAE*
Total	5.1802	3.3152	6.7106	4.2806
Hudson	8.2871	6.0331	11.5791	8.8327
Manhattan	4.4532	2.7495	5.2845	3.1740
Brooklyn	5.2661	3.4681	6.8584	4.6273
Bronx	7.6168	5.5544	11.2118	8.5488
Queens	6.0233	4.0720	8.2514	5.7618

good. Sometimes, even our CT methods can achieve very similar results then HAM without using any historical data on the predicted areas, such as "Manhattan" → "Queens" in the last sub-figure. Apart from that, we found that a larger region like "Manhattan" which contains various kinds of links will have better CT performance than other regions. Whereas, neural network model is relatively unstable. As for different prediction models, we conclude that our proposed CTPM performs substantially better than other classic feature-based methods.

References

Box, G. 2013. *Box and Jenkins: Time Series Analysis, Forecasting and Control*. Palgrave Macmillan UK.

Clark, S. 2003. Traffic prediction using multivariate nonparametric regression. *Journal of Transportation Engineering* 129(2):161–168.

Keogh, E. J., and Pazzani, M. J. 2001. Derivative dynamic time warping. In *SDM*.

Lin, L.; Li, J.; Chen, F.; Ye, J.; and Huai, J.-P. 2017. Road traffic speed prediction: A probabilistic model fusing multi-source data. *IEEE TKDE*.

Mchugh, D. 2015. Traffic Prediction and Analysis using a Big Data and Visualisation Approach.

Pan, S. J., and Yang, Q. 2010. A survey on transfer learning. *IEEE TKDE* 22(10):1345–1359.

Pan, B.; Demiryurek, U.; and Shahabi, C. 2012. Utilizing real-world transportation data for accurate traffic prediction. *Proceedings - IEEE International Conference on Data Mining, ICDM* 595–604.

Ren, Y.; Ercsey-Ravasz, M.; Wang, P.; González, M. C.; and Toroczkai, Z. 2014. Predicting commuter flows in spatial networks using a radiation model based on temporal ranges. *Nature communications* 5:5347.

Xu, J.; Deng, D.; Demiryurek, U.; Shahabi, C.; and Schaar, M. V. D. 2015. Context-aware online spatiotemporal traffic prediction. *IEEE International Conference on Data Mining Workshops, ICDMW* 2015-Janua(January):43–46.

Xu, F. F.; Lin, B. Y.; Lu, Q.; Huang, Y.; and Zhu, K. Q. 2016. Cross-region traffic prediction for china on openstreetmap. In *Proceedings of the 9th ACM SIGSPATIAL International Workshop on Computational Transportation Science*, 37–42. ACM.

Yao, B.; Chen, C.; Cao, Q.; Jin, L.; Zhang, M.; Zhu, H.; and Yu, B. 2017. Short-term traffic speed prediction for an urban corridor. *Computer-Aided Civil and Infrastructure Engineering* 32(2):154–169.

Zheng, Y.; Capra, L.; Wolfson, O.; and Yang, H. 2014. Urban computing: concepts, methodologies, and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)* 5(3):38.