Abstract

Traffic speed prediction can benefit a wide range of downstream applications of machine learning for intelligent transportation systems. Most existing approaches leverage historical time-series data, which thus makes them only able to work on the areas where large amount of historical data is available. To predict traffic speed of target areas where little historical traffic data is available, we propose a transfer learning framework that exploits historical data of some source areas and various effective spatiotemporal features. Experimental results show that classic regression models and our proposed novel model can achieve competitive performance comparing to baseline methods which utilize historical data of target areas.

1 Introduction

Traffic speed prediction is a challenging problem and has various downstream machine learning applications, many of which are fundamental to intelligent transportation systems, such as congestion management, transportation planning, vehicle routing, etc. [5, 10, 4] Most of the existing approaches heavily rely on the historical data of the areas being predicted [7, 2]. However, accurate and reliable historical traffic data collected from road sensors is very expensive and available in only a few areas, where the government can afford the large cost. Consequently, most state-of-the-art time-series based models cannot be applied easily on those areas where little historical traffic data is available.

Another problem of most existing models is that they only utilize temporal features and do not capture the relationship between spatial features and temporal patterns [11, 3]. Such knowledge can benefit many practical applications of urban computing [12], but research on extracting effective spatial features for traffic speed prediction is almost missing from the literature.

In this paper, we aim to answer this research questions: How can we exploit the existing historical traffic speed data of some areas to make speed predictions for other areas without their data?

We attack this problem by proposing a feature-based transfer learning [8] approach that exploits both historical traffic speed data of other areas and effective spatial features. Our contribution in this paper is as follows: 1) The proposed transfer learning approach supports many classic regression models. 2) We extract various spatial features in multiple levels and combine them with temporal features to support the transfer learning scenario and improve interpretability of the proposed model. 3) Experimental results show that proposed transfer learning models based on our effective spatiotemporal features can perform competitively with two classic regression models for predicting traffic speed. 4) To the best of our knowledge, we are among the first to study transfer learning for traffic speed prediction [8].

*The first two authors contribute equally.
2 Problem Statement

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. A transfer learning approach is supposed to exploit the data of some source areas $S$ to build a prediction model for other target areas $T$ of interest, where there is little traffic speed data. Mathematically, given $\{v_i[1 : t]|i \in S\}$, a transfer learning model should be able to predict $\{v_j[1 : t]|j \in T\}$.

3 Data

In this section, we briefly introduce a public dataset named UIUC New York City Traffic Estimates, 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. Most importantly, 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; 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 areas as shown in Table 1.

<table>
<thead>
<tr>
<th>#link</th>
<th>Hudson</th>
<th>Manhattan</th>
<th>Brooklyn</th>
<th>Bronx</th>
<th>Queens</th>
</tr>
</thead>
<tbody>
<tr>
<td>#trips_sum</td>
<td>730</td>
<td>8,378</td>
<td>7,790</td>
<td>2,113</td>
<td>8,173</td>
</tr>
<tr>
<td>477,596</td>
<td>52,394,074</td>
<td>24,121,488</td>
<td>5,002,542</td>
<td>20,531,664</td>
<td></td>
</tr>
</tbody>
</table>

4 Spatiotemporal Features

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

4.1 Basic Information Features

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 link length is the real distance between the begin and the end node of this link. The number of in and out links connected to both nodes determines the number of in and out links of the two end nodes accordingly.

4.2 Road Density Features

Additionally, we believe traffic speed is highly relevant to road density, which can be measured by the number 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). Consequently, we have $2 \times 3 \times 3 = 18$ road density features in total.

---

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

[https://www.openstreetmap.com](https://www.openstreetmap.com)
4.3 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 for each node with three different radius (100/300/500m) using HERE Places API. The POI types we consider are shown in Table 2. Figure 1 shows such an example for extract road density features and POI density features.

Table 2: 11 Main-types of POI

<table>
<thead>
<tr>
<th>Eat &amp; Drink</th>
<th>Going Out</th>
<th>Sights &amp; Museums</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transport</td>
<td>Accommodation</td>
<td>Shopping Facility</td>
</tr>
<tr>
<td>Business &amp; Services</td>
<td>Facilities</td>
<td>Natural or Geographical</td>
</tr>
<tr>
<td>Administrative Areas/Buildings</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 1: An Example of Extracting POI Features. Different colors indicate different POI types.

4.4 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.

5 Proposed Feature-based Transfer Learning Models

Obtaining the above features for link and time, we first apply several classic machine learning models for regression that are trained on source areas with above-explored spatiotemporal features and then simply predict traffic condition on target areas as a test. Afterwards, we present a novel transfer learning approach named CTMP.

Notations: a prediction query instance about a link \( l \) at the time \( t \) is denoted as \((l, t)\); the spatial feature vector for the link \( l \) is denoted as \( s_l \), and the temporal feature vector of the time \( t \) being predicted on the link \( l \) is denoted as \( t_l \). The ground truth of the speed is denoted as \( v_l[t] \).

5.1 Linear Regression Model

Linear regression (LR) is an approach for modeling the relationship between a scalar dependent variable \( y \) and one or more explanatory variables denoted \( x \). The model can be expressed in the

\[ y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n + \epsilon \]

where \( \beta_i \) are the coefficients, and \( \epsilon \) is the error term.
following form: \( y = \mathbf{w}^T \mathbf{x} + b \), where \( \mathbf{w} \) and \( b \) are the parameters we should learn in order to optimize a particular loss. In our case, we regard the \( \mathbf{x} \) for each query instance as the concatenation of the spatial feature \( s_l \) and the temporal feature \( t_l \). Thus, we have \( \mathbf{x} = [s_l; t_l] \).

5.2 Neural Network Model

We adopt a four-layer neural network model (MLP) with following structure: the size of input layer is equal to the dimensionality of \( \mathbf{x} \); the second layer contains half of it; the third is half of the second layer; the final output layer contains only one neuron. The model (NN) takes the same input as before and output a continuous value by the last output layer which is the predicted speed.

5.3 Support Vector Regression Model

Support vector regression (SVR) depends only on a subset of the training data, because the cost function for building the model ignores any training data close to the model prediction. Training the original SVR means solving following optimization problem, where \( x_i \) is a training sample with target value \( y_i \):

\[
\min \frac{1}{2} \| \mathbf{w} \|^2, \quad \text{subject to} \quad \begin{cases} 
& y_i - \langle \mathbf{w}, x_i \rangle - b \leq \varepsilon \\
& \langle \mathbf{w}, x_i \rangle + b - y_i \leq \varepsilon 
\end{cases}
\]

5.4 CTMP: A Two-stage Clustering-based Transfer Model

We introduce our novel Clustering-based Transfer Model for Prediction (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.

5.4.1 Intuition Behind the CTMP

Our intuition behind CTMP is that given a link in target areas with its 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. Our basic assumption 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.

5.4.2 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 choose to capture the distance between the two feature vectors by computing

\[
s_{\text{dis}}(i, j) = 1 - \cos(s_i, s_j).
\]

To regularize the time series similarities between two links, we add a regularization term \( t_{\text{dis}}(i, j) \), which has multiple options. A desirable option is the DTW 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 \( \lambda \) is a hyper parameter to control the weight of temporal distance:

\[
d_{\text{dis}}(i, j) = s_{\text{dis}}(i, j) + \lambda t_{\text{dis}}(i, j)
\]

With such supervision in the source area data, we can use K-means as our clustering algorithm. For each query instance \((l, t)\)\(^5\), we first find the closest \( k \) neighboring source links with historical data \( \{l_1, \ldots, l_k\} \). We compute all the distances between them and the target link \( l \) respectively, and obtain the set of spatial feature distances

\[
\{d_{\text{dis}}(l, l_1), \ldots, d_{\text{dis}}(l, l_k)\}.
\]

\(^5\) Typically, the link \( l \) has no available historical data.
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:
\[
y(l, t) = \sum_{i=1}^{k} \left( \frac{\text{dis}(l, l_i)}{\sum_{j=1}^{k} \text{dis}(l, l_j)} y(l_i, t) \right)
\]

6 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.

6.1 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 (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.

6.2 Baseline Methods

To compare our model with the most common time-series based regression model, we choose two representative ones: ARIMA and HAM.

6.2.1 Auto-Regressive Integrated Moving Average (ARIMA)

ARIMA models are applied on some cases where time series data shows evidence of non-stationarity and an initial differencing step (corresponding to the "integrated" part of the model) can be applied one or more times to eliminate the non-stationarity. Prediction of the data for the time \( t + 1 \) is:
\[
Y_{t+1} = \sum_{i=1}^{p} \alpha_i Y_{t-i+1} + \sum_{i=1}^{q} \beta_i \epsilon_{t-i+1} + \epsilon_{t+1}
\]
where \( Y_t \) refers to a time series data. In the autoregressive component of this model, a linear weighted combination of previous data is calculated, where \( p \) refers to the order of this model and \( \alpha_i \) refers to the weight of \((t - i + 1)\)-th reading. As for the second part, the sum of weighted noise from the moving average model is calculated, where \( \epsilon \) denotes the noise, \( q \) refers to its order and \( \beta_i \) represents the weight of \((t - i + 1)\)-th noise. In our experiment, we trained ARIMA on the first half year to pick the parameters with best performance using 5-fold cross-validation.

6.2.2 Historical Average Model (HAM)

HAM utilizes the average of previous speed data for the same time and same location to forecast the future data. HAM is formulated as follows:
\[
v(t_d, w + h) = \frac{1}{|V(d, w)|} \sum_{s \in V(d, w)} v(s)
\]
where \( V(d, w) \) refers to the subset of past observations that happened at the same time \( d \) on the same weekday \( w \). Specifically, \( d \) captures the daily effects while \( w \) captures the weekly effects. 7 * 24 average regular speed tables for each link are derived from the data of first half year. The model is evaluated by comparing using such tables.
6.3 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 3. We found that HAM performs substantially better than the ARIMA model with both metrics, which is probably due to the time interval in our data is one hour, quite longer than common time interval length for ARIMA model.

<table>
<thead>
<tr>
<th></th>
<th>HAM RMSE</th>
<th>HAM MAE</th>
<th>ARIMA RMSE</th>
<th>ARIMA MAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total</td>
<td>5.1802</td>
<td>3.3152</td>
<td>6.7106</td>
<td>4.2806</td>
</tr>
<tr>
<td>Hudson</td>
<td>8.2871</td>
<td>6.0331</td>
<td>11.5791</td>
<td>8.8327</td>
</tr>
<tr>
<td>Manhattan</td>
<td>4.4532</td>
<td>2.7495</td>
<td>5.2845</td>
<td>3.1740</td>
</tr>
<tr>
<td>Brooklyn</td>
<td>5.2661</td>
<td>3.4681</td>
<td>6.8584</td>
<td>4.6273</td>
</tr>
<tr>
<td>Bronx</td>
<td>7.6168</td>
<td>5.5544</td>
<td>11.2118</td>
<td>8.5488</td>
</tr>
<tr>
<td>Queens</td>
<td>6.0233</td>
<td>4.0720</td>
<td>8.2514</td>
<td>5.7618</td>
</tr>
</tbody>
</table>

Figure 2: Local Transfer and Cross-region Transfer Performances in terms of RMSE; Each sub-figure is about region, and each cluster is either a Local Transfer(LT) or a Cross-region Transfer(CT).

Also, we present the results of both Local Transfer (LT) and Cross-region Transfer (CT) in Figure 2. 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 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. As for different prediction models, we found that our proposed CTPM performs better than other classic feature-based methods. Whereas, NN model is relatively unstable.

7 Conclusion

In this paper, we study a both challenging and significant scenario of traffic prediction, which is to predict the future speed of a given road link but without its historical data. We first extracted various spatial features to represent the geographical information of each links together with their temporal features, and then we design several feature-based transfer prediction models. Also, we proposed a novel two stage prediction model which clusters the links by their spatial features with regularization of their temporal similarities.
References


