Artificial Intelligence in Transportation

Zheng WANG
DiDi AI Labs
Didi Chuxing

Yan LIU
DiDi AI Labs
Univ. of Southern California

Jieping YE
DiDi AI Labs
Univ. of Michigan, Ann Arbor
Outline

- Challenges and opportunities in transportation AI (20min)
  - Overview of urban transportation
  - The emerging challenges in transportation AI

- AI applications in transportation (165min+Break)
  - Map services I: map matching, route planning, estimated time of arrival (ETA) (60min)
  - Break (30min)
  - Map services II: traffic estimation, traffic forecast (45min)
  - Decision making services: dispatching (30min)
  - AI applications and AI for social good (30min)

- Data and tools for transportation AI (15min)

- Q&A (10min)
Part 1: Challenges and Opportunities in Transportation AI
History of Urban Transportation

**First Wheeled Vehicle**: 3500 B.C.

**First Road Network**: 1st Century

**First Public Transportation**: 1662

**First Bicycle**: 1817

**First Affordable Automobile**: 1908

**First Electric Traffic Light**: 1910s

**U.S. National Highway System**: 1956

**Smart Transportation System**: Now and Future
History of Traffic Light

**Before 1860s**
- No Traffic Light
- Hand-operated Traffic Light

**1910s**
- Electric Traffic Light

**1920s**
- Weigh in Motion
- Camera
- Loop Detector
- Radar
- Weigh in Motion

**1980s**
- Camera
- Loop Detector
- Weigh in Motion
- Radar
Transportation: A Multi-Disciplinary Industry
Cutting-Edge Issues

Cooperative Vehicle-Highway Systems

Ride Sharing

Multimodal Transportation

- microtransit
- car sharing
- bike sharing
- ride sourcing
- ride sharing
- public transit
- taxi
Smart Transportation System

**Smart Travelers**
From Drive Alone to Ride Sharing

**Smart Vehicles**
From Human Driving to Autonomous Driving

**Smart Infrastructure**
From Independent Systems to Cooperative Vehicle-Highway Systems

Cloud

Big Data

Transportation Engineering

AI
Building the Brains of Smart Transportation

Intelligent and Connected Vehicles

Future Intersection
AI and Machine Learning

Neural Networks

Deep Learning

Machine Learning: supervised, unsupervised

Reinforcement Learning

AlphaGo
Data Resource

- Location data, Trajectory data
- Transaction data
- Profile data
- Sensors: multimedia data
- Cross-platform identification
GPS Data

Location Data and Floating-Car Trajectory
Sensors

Loop detector, camera, microphone, mobile sensors ...
Big data makes AI possible for transportation.
Smart Transportation Brain

- Route Planning
- ETA
- Pick-up locations
- VR Navigation
- Demand-Supply Prediction
- Order Dispatch
- Car Pooling
- Resource Allocation
- Multi-modal
- Taxi
- Express
- Car Pool
- Premiere
- Ride-sharing Services
- Map Services
- Platform Optimization
- Analysis
- Data Collection
- Control
- Signal Control
- Freeway Control
- Traffic Guidance
- Incident Management
- AI Dispatch
- Performance Measures
- Congestion Diagnosis
- Network Design
- Traffic Simulation
- Accident Analysis
- DiDi Data
- Government Data
- Collaborators’ Data
- Crowd Sourced Data
Outline

- **Challenges and opportunities in transportation AI (20min)**
  - Overview of urban transportation
  - The emerging challenges in transportation AI

- **AI applications in transportation (165min+Break)**
  - Map services I: map matching, route planning, estimated time of arrival (ETA) (60min)
  - Break (30min)
  - Map services II: traffic estimation, traffic forecast (45min)
  - Decision making services: dispatching (30min)
  - AI applications and AI for social good (30min)

- **Data and tools for transportation AI (15min)**

- **Q&A (10min)**
Part 2: AI Applications in Transportation
Future Transportation

Smart Infrastructure

Smart Vehicle

Shared Mobility

Highway, Road, Smart Traffic Light, ...

Electrical Vehicle, Autonomous Vehicle, ...

Map Service, Decision Service, ...
Key Components

- Basic Layer: map service and LBS
- Upper Layer: decision service and marketplace
Map Service I
Smart Map for Modern Transportation

- Navigation
- Ride Hailing
- Transportation System Efficiency
- Autonomous Driving
Core Map Service

Route Planning

ETA

Map Matching

Positioning

Traffic
Map Matching
Map Matching

• Problem Definition

• Solutions to Map Matching

• Challenges
Map Matching

Road Network

GPS Points
Naive Approach: Nearest Neighbor

Road Network

GPS Points
• Model this process using Hidden Markov Model.

Emission Probability

GPS Deviation Follows Gaussian Distribution

\[ p(z_t | r_i) = \frac{1}{\sqrt{2\pi \sigma_z}} e^{-0.5 \left( \frac{\|z_t - x_{t,i}\|_{gcourse}}{\sigma_z} \right)^2} \]

Distance Between GPS and Matched Point (meters)

* Part of this slide is borrowed from: Paul Newson et al. Hidden Markov Map Matching Through Noise and Sparseness. ACM SIGSPATIAL 2009
Transition Probability

\[ p(d_t) = \frac{1}{\beta} e^{-d_t / \beta} \]

Distance Difference Distribution

\[ d_t = \| z_t - z_{t+1} \|_{\text{great circle}} - \| x_{t,i}^* - x_{t+1,j}^* \|_{\text{route}} \]

* Part of this slide is borrowed from: Paul Newson et al. Hidden Markov Map Matching Through Noise and Sparseness. ACM SIGSPATIAL 2009
Parameter Estimation

• In reality: heuristic estimation
  • Emission probability parameter (noise in location measurements):
    \[ \sigma_z = 1.4826 \text{median}_t \left( \| z_t - x_{t,i^*} \|_{\text{great } \text{circle}} \right) \]
  • Transition probability parameter (tolerance of non-direct routes):
    \[ \beta = \frac{1}{\ln(2)} \text{median}_t \left( \| z_t - z_{t+1} \|_{\text{great circle}} \right. \\
    - \left. \| x_{t,i^*} - x_{t+1,j^*} \|_{\text{route}} \right) \]
• Parameter refinement
• In literature: parameter learning
Parameter Learning with IRL

- Map Matching with Inverse Reinforcement Learning
  - Transition probability variant, HMM
  - \( P_{\text{trans}} = \frac{1}{\beta} \exp\left( -\frac{|r^* - g_0|}{\beta} \right) \)
  - Conventionally, \( r^* = \|d_1 + d_2 + d_3 + d_+ - d_- \| \)
  - Here \( r^* = \|d_1 + d_2 + d_3 + d_+ - d_- + w_{\text{turn}}(u_{01} + u_{12} + u_{23}) \| \)

\[
 u_{v,v'} = \begin{cases} 
 0 & \text{if } |\theta_{v,v'}| < \pi/4, \\
 1 & \text{if } \pi/4 \leq |\theta_{v,v'}| \leq 3\pi/4, \\
 2 & \text{if } 3\pi/4 < |\theta_{v,v'}| < \pi, \\
 10 & \text{if } |\theta_{v,v'}| = \pi.
\end{cases}
\]

- Weight estimation \( w_{\text{turn}} \) with IRL

State Estimation using Viterbi Algorithm

• Map-matching as state estimation
  • Input: a sequence of GPS points, HMM
  • Output: the most likely state sequence, i.e., a sequence of edges in the road network.

• Viterbi Algorithm
  • Dynamic Programming based method to identify the state sequence with the highest probability.
Challenges

• Large-scale GPS data

• Low-quality GPS data for mobile device

• Limited amount of labeled data
  • Unsupervised learning: EM for HMM
  • Semi-supervised learning
Reference

• Sotiris Brakatsoula et al. On map-matching vehicle tracking data, VLDB 2005
• Paul Newson et al. Hidden Markov map matching through noise and sparseness, ACM SIGSPATIAL 2009
• Yin Lou et al. Map-matching for low-sampling-rate GPS trajectories, ACM SIGSPATIAL 2009
• T. Osogami et al. Map Matching with Inverse Reinforcement Learning, IJCAI 2013
Route Planning
Route Planning
Route Planning

• Challenges:
  • Large-scale transportation network
  • High query speed
  • Accurate result

• Classical problem: shortest path algorithm
  • Dijkstra’s algorithm and its extensions: (high) query speed, less robust
  • A*-algorithm: robust, low query speed
  • Customizable routing: robust, relative high query speed

• Data-driven approaches
  • What is the proper edge weight for the graph?
  • How can we take advantage of the big data?
Shortest Path

• Build a graph based on map and traffic information.
• Graph edge weight for travel cost
• Find a route with the minimum travel cost.
Dijkstra's algorithm: a greedy approach

• Given a weighted graph with nonnegative edge weights, Dijkstra's algorithm finds the shortest path between nodes in the graph.

The complexity of Dijkstra’s algorithm is $O(|e| + |v|\log |v|)$. 
A*-Algorithm: a heuristic approach

- Search with a heuristic guidance

\[ f(n) = g(n) + h(n) \]

Speedup

- Speedup Dijkstra’s Algorithm

Dijkstra’s Algorithm

Bidirectional Dijkstra’s Algorithm

Graph Contraction
Fast Shortest Path

• Contraction Hierarchies (CH)
  • Preprocessing: sorting the nodes, adding shortcut, layer-wise contraction
  • Query: bidirectional Dijkstra's algorithm, unfolding the shortcut

**Contraction Hierarchies** method is a technique to speed up shortest path routing by first creating precomputed "contracted" versions of the connection graph. It can be regarded as a special case of "highway-node routing".
Component Reduction

- Remove nodes and edges w/o shortcut.
Order for Contraction

• Setting the node order for contraction: edge difference, cost of contraction, uniformity, cost of queries ...
• Order for contration and shortest path search.
Graph Contraction

- Multi-level contraction
Bidirectional Search

• Bidirectional Dijkstra's algorithm: forward search goes from low order node to high order node, and vice versa.
• Unpack the shortest path.
Guarantee

- Proved by contradiction
Multiple Metric

• CRP: customizable route planning
  • Metric independent processing (partition)
  • Metric customization
  • Query
Partition

- Multi-layer partition on unweighted graph
- Overlay graph: distance preserving subgraph
• Three possible ways of preserving distances within the overlay graph: full clique, arc reduction and skeleton.

* Picture from Daniel Delling et al. *Customizable Route Planning in Road Networks*. Transportation Science 2017.
Customization

Pruning

* Picture from Daniel Delling et al. *Customizable Route Planning in Road Networks*. Transportation Science 2017.
Query

• Shortest path in a subgraph: overlay graph + subgraph including OD nodes.

• Unpack the shortest path: Dijkstra’s Algorithm for each clique.

* Picture from Daniel Delling et al. *Customizable Route Planning in Road Networks*. Transportation Science 2017.
Summary of Shortest Path Search

Results on road network of West Europe, using travel times as edge weights. (18M nodes and 42.5M edges)

Challenges

Travel Distance  Travel Speed  Traffic condition  Driver/Passenger Preference  Data Driven vs Heuristic

It is very difficult to preset proper penalty weight to represent all those factors.
Can we learn human driving patterns?
Data Driven: decision process

Approximately, Route Planning problem can be seen as a deterministic MDP problem. Our goal is to maximize/minimize customer’s reward/cost.
Data Driven

Classification / Reinforcement learning (driving policy) problem
Data Driven

1. Selection

How to guarantee the long time planning effectiveness?
Data Driven

2. Expand & Evaluate
Data Driven

3. Backtrack
4. Execute

\[ L = \sum_{t} (v(s_t; w) - z_t)^2 - \dot{\pi}_i \log \pi(a|s_t; \theta) \]
\[ + c||\theta||^2 + c||w||^2 \]
Learning to Plan the Route

• Modeling Trajectories with Recurrent Neural Networks

• Route Planning with Reinforcement Learning
Modeling Trajectories with Recurrent Neural Networks

• RNN: capture variable length sequence

• Similarity & difference in language/trajectory modeling using RNN
  • Similar: generate words/edges step by step, depending on the present and past words/edges.
  • Different: the transition from one word to any other word is free, while only the transition from one edge to its adjacent edges is possible.
Modeling Trajectories with Recurrent Neural Networks

• Goal: use RNN to learn the topological constraints.

• Solution: modification of state-constrained softmax function.

\[
p(\tilde{r}_{t+1} | r_{1:t}) = C(Wh_t + b, r_t) = \frac{\exp (Wh_t + b) \odot M_i}{\| \exp (Wh_t + b) \odot M_i \|_1}
\]

\[
M_{ij} = \begin{cases} 
1 & \text{if } r_i \text{ can reach } r_j \\
0 & \text{otherwise}
\end{cases}
\]
Modeling Trajectories with Recurrent Neural Networks
## Modeling Trajectories with Recurrent Neural Networks

<table>
<thead>
<tr>
<th>Task</th>
<th>With Destination</th>
<th>PT\textsubscript{small}</th>
<th>PT\textsubscript{large}</th>
<th>SH\textsubscript{small}</th>
<th>SH\textsubscript{large}</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>NLL</td>
<td>ACC</td>
<td>NLL</td>
<td>ACC</td>
</tr>
<tr>
<td>Dataset</td>
<td></td>
<td>NLL</td>
<td>ACC</td>
<td>NLL</td>
<td>ACC</td>
</tr>
<tr>
<td>Metric</td>
<td></td>
<td>NLL</td>
<td>ACC</td>
<td>NLL</td>
<td>ACC</td>
</tr>
<tr>
<td>Bi-gram</td>
<td></td>
<td>6.20</td>
<td>94.15%</td>
<td>8.91</td>
<td>92.30%</td>
</tr>
<tr>
<td>Tri-gram</td>
<td></td>
<td>6.20</td>
<td>93.88%</td>
<td>8.92</td>
<td>91.99%</td>
</tr>
<tr>
<td>4-gram</td>
<td></td>
<td>6.21</td>
<td>93.57%</td>
<td>8.93</td>
<td>91.66%</td>
</tr>
<tr>
<td>BIRL</td>
<td></td>
<td>5.84</td>
<td>95.53%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MERIL</td>
<td></td>
<td>7.84</td>
<td>93.70%</td>
<td>8.87</td>
<td>93.23%</td>
</tr>
<tr>
<td>RNN</td>
<td></td>
<td>3.74</td>
<td>97.13%</td>
<td>5.63</td>
<td>96.65%</td>
</tr>
<tr>
<td>CSSRNN</td>
<td></td>
<td>3.21</td>
<td>97.16%</td>
<td>3.96</td>
<td>96.89%</td>
</tr>
<tr>
<td>LPIRNN</td>
<td></td>
<td>3.12</td>
<td>97.21%</td>
<td>3.98</td>
<td>96.97%</td>
</tr>
</tbody>
</table>
Imagination-Augmented Agent (I2A), which incorporating model-free RL and model-based RL, improves data efficiency, performance, and robustness.
Route Planning with Reinforcement Learning

Model-Based Approaches

**Advantages**
- Endowing agents with a model of the world
- Support generalization to states not previously experienced

**Disadvantages**
- Complex domains hard to build environment models.
- Performance suffers from model errors resulting.

Model-Free Approaches

**Disadvantages**
- Requires large amounts of training data
- Resulting policies do not readily generalize to novel tasks in the same environment
Route Planning with Reinforcement Learning

Key of Model-Based method —— Environment Model

- **Input**: Observed State, Chosen Action → **Output**: Next State, Reward
- **State**
- **Action**
- **Reward**

**Environment Model Construction Options**

- Make **assumptions** about the structure of the environment model with domain knowledge
- Trained directly on low-level observations with little domain knowledge, similarly to recent model-free successes. *(e.g., I2A)*
- **Perfect** World Model
Route Planning with Reinforcement Learning

Output Layer of Model-based: Path Encoder

Role:
EM generate path after taking a specific action via imagination.
Encoder transform the path into the evaluation of selected action.

Realized by RNN

Imagination Path

Action: Left
Route Planning with Reinforcement Learning

- I2A performs better than others
- Performance $\propto$ Imagination Steps
- Diminishing returns with more rollout steps
Route Planning with Reinforcement Learning

Address the challenge

• RNN Encoder captures the sequential information.
• Work well even with imperfect EM (learn to ignore the latter as errors accumulate)
Alternative Solution

• Learning to navigate in cities without a map, from deep mind.


ETA (Estimated Time of Arrival)
ETA (Estimated Time of Arrival)
Main Stream ETA Models

• Additive models:
  • Rule-based additive models: explicitly modeling the segments in a path
  • Aggregating the time of sub-paths
  • Machine learning models for the sub-path problem.

• Global models:
  • Formulating ETA as a regression problem: Learning to Estimate the Travel Times (L2ETT)
  • Simple regression model and deep learning model

• Path-free models:
  • Path is not available
Simple Additive Model

- Simple rules based on physical structure of road network
- Popular solution in digital map industry
- Challenges: precise speed estimation and error accumulation

\[ ETA = \sum_{i=1}^{n} t_i + \sum_{j=1}^{m} c_j \]
Aggregating Sub-Paths

- The path can be decomposed into sub-paths.
- Time of each sub-path is inferenced from the historical trajectories.
- ETA is the summation of the time of sub-paths.

\[ N(q) = \{ p_i \in D | dist(o_i, o_q) \leq \tau \text{ and } dist(d_i, d_q) \leq \tau \}, \quad \hat{t}_q = \frac{1}{|N(q)|} \sum_{p_i \in N(q)} w_i t_i. \]

* Hongjian Wang et al. A simple baseline for travel time estimation using large-scale trip data. SIGSPATIAL 2016.
Tensor Decomposition

- Build a 3D tensor (driver, road segment, time slot) and use tensor decomposition to estimate the travel time of each segment.
- Use dynamic programming to find the optimal concatenations.

* Yilun Wang et al. Travel time estimation of a path using sparse trajectories. KDD 2014.
Regression Model on GPS Points

- Deep regression model on GPS data
  - GPS points will not be available before trip in real-world case.

Time Series Prediction

- Conventional regression model: support vector regression
- Deep learning model: recurrent neural network

\[ f(t-n),\ldots,f(t-1) \rightarrow f(t) \]
\[ f(t-n),\ldots,f(t-m) \rightarrow f(t) \]

Probabilistic ETA

• Output a distribution of ETA
• Consider the variance of travel time
• Assume Gaussian distribution for link travel time

Additive Model vs Global Model

- 1 Heuristic rule
- 2 Indirective objective
- 3 Less robust
- 4 Insufficient use of data

Additive model?

Pure learning?
Rethinking ETA

Historical & real-time traffic data → Segment-wise statistic and accumulation → Direct estimation by regression model → Statistical machine learning
Learning to Estimate the Travel Time

- Big data + machine learning
  - high accuracy
  - data driven
  - robustness

Diagram:
- Raw data
- Feature extraction
- Static/dynamic feature
  - Dense feature
  - Sparse feature
- Machine learning model
- ETA service
Problem Formulation

• Formulation: ETA as a regression problem on sequential input

Data: a collection of trips $X = \{x_i\}_{i=1}^N$
  • $x_i$ denotes feature vector on the trajectory along the route path $p_i$
  • more than 20 million trips a day

Objective: for MAPE loss, $\min_f \sum_{i=1}^N \frac{|y_i - f(x_i)|}{y_i}$

Robust: different training data for different scenarios
  • picking the passenger
  • delivering the passenger
  • order dispatch
  • car pooling
Two faces: global feature vs sequential feature

Raw data

Raw data

Feature

Label $T_{od}$
Recurrent Model

- Model sequential features with Recurrent Neural Network (RNN).
Tree-Based Model -> Wide & Deep Model

- **Features**
  - Spatial information
  - Temporal information
  - Traffic information
  - Personalized information
  - Augmented information

- **W&D model processes the global information**
Wide-Deep-Recurrent Network (WDR)

## Offline Evaluation Dataset

<table>
<thead>
<tr>
<th></th>
<th>date</th>
<th>pickup</th>
<th>trip</th>
</tr>
</thead>
<tbody>
<tr>
<td>training set</td>
<td>1.1 - 5.10 (2017)</td>
<td>48M</td>
<td>51M</td>
</tr>
<tr>
<td>validation set</td>
<td>5.11 - 5.17 (2017)</td>
<td>3M</td>
<td>4M</td>
</tr>
<tr>
<td>test set</td>
<td>5.18 - 5.31 (2017)</td>
<td>6M</td>
<td>7M</td>
</tr>
<tr>
<td>unique links</td>
<td>–</td>
<td>0.5M</td>
<td>0.5M</td>
</tr>
<tr>
<td>unique drivers</td>
<td>–</td>
<td>0.4M</td>
<td>0.4M</td>
</tr>
</tbody>
</table>
## Offline Evaluation Results

<table>
<thead>
<tr>
<th></th>
<th>MAPE</th>
<th>MAE</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>route-ETA</td>
<td>31.27%</td>
<td>88.0</td>
<td>16555.8</td>
</tr>
<tr>
<td>PTTE [20]</td>
<td>29.35%</td>
<td>83.3</td>
<td>13153.5</td>
</tr>
<tr>
<td>GBDT</td>
<td>23.64%</td>
<td>68.3</td>
<td>11674.2</td>
</tr>
<tr>
<td>FM</td>
<td>21.41%</td>
<td>63.6</td>
<td>9664.2</td>
</tr>
<tr>
<td>WD-MLP</td>
<td>21.58%</td>
<td>64.1</td>
<td>9816.3</td>
</tr>
<tr>
<td>WDR</td>
<td><strong>20.83%</strong></td>
<td><strong>59.9</strong></td>
<td><strong>9078.2</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>MAPE</th>
<th>MAE</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>route-ETA</td>
<td>15.01%</td>
<td>153.2</td>
<td>66789.8</td>
</tr>
<tr>
<td>PTTE [20]</td>
<td>14.78%</td>
<td>150.1</td>
<td>66141.2</td>
</tr>
<tr>
<td>GBDT</td>
<td>14.01%</td>
<td>133.2</td>
<td>52006.6</td>
</tr>
<tr>
<td>FM</td>
<td>12.87%</td>
<td>122.5</td>
<td>47457.3</td>
</tr>
<tr>
<td>WD-MLP</td>
<td>13.43%</td>
<td>127.5</td>
<td>50012.8</td>
</tr>
<tr>
<td>WDR</td>
<td><strong>11.66%</strong></td>
<td><strong>112.4</strong></td>
<td><strong>37482.7</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>MAPE</th>
<th>MAE</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>*TEMP&lt;sub&gt;rel&lt;/sub&gt; [19]</td>
<td>35.30%</td>
<td>79.7</td>
<td>15480.7</td>
</tr>
<tr>
<td>*WDR</td>
<td><strong>21.68%</strong></td>
<td><strong>55.6</strong></td>
<td><strong>7962.0</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>MAPE</th>
<th>MAE</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>*TEMP&lt;sub&gt;rel&lt;/sub&gt; [19]</td>
<td>23.53%</td>
<td>166.8</td>
<td>88767.1</td>
</tr>
<tr>
<td>*WDR</td>
<td><strong>12.27%</strong></td>
<td><strong>87.2</strong></td>
<td><strong>20929.3</strong></td>
</tr>
</tbody>
</table>
WDR achieves the best performance for all metrics.
Progress

Regression Model
- Low risk of overfitting
- Easy to implement
- Fast training and inference

Deep Learning Model
- Better interpretability
- Better performance
- Sequential information

Model Compression
- High inference speed
- More portable
More Practical: Origin Destination ETA

Route is not available until the end of the trip.
Origin Destination ETA

- Origin-destination ETA (path free)

Challenges
- Limited information: no actual path
- Complicated spatiotemporal dependencies
MURAT: Multi-task Representation Learning for ETA

Road Network Embedding

Spatiotemporal Smoothness
MURAT: Multi-task Representation Learning for ETA

Historical Paths

- Travel Distance
- # Road Segments
- # Traffic lights
- # Left/right Turns

Auxiliary Tasks
MURAT: Multi-task Representation Learning for ETA

* Yaguang Li et al. Multi-task Representation Learning for Travel Time Estimation, KDD, 2018
Evaluation

Beijing
60M+ trips

New York City
20M+ trips

Mean Absolute Relative Error

<table>
<thead>
<tr>
<th>Model</th>
<th>Beijing 60M+ trips</th>
<th>New York City 20M+ trips</th>
</tr>
</thead>
<tbody>
<tr>
<td>LR</td>
<td>37.7%</td>
<td>32.4%</td>
</tr>
<tr>
<td>GBM</td>
<td>33.5%</td>
<td>27.0%</td>
</tr>
<tr>
<td>ST-NN</td>
<td>32.5%</td>
<td>22.6%</td>
</tr>
<tr>
<td>MURAT-NR</td>
<td>33.4%</td>
<td>22.1%</td>
</tr>
<tr>
<td>TEMP+R</td>
<td>29.4%</td>
<td>22.1%</td>
</tr>
<tr>
<td>MURAT</td>
<td>21.0%</td>
<td>21.0%</td>
</tr>
</tbody>
</table>
Learned Representation

Circular Shape

Smooth transition
• Erik Jenelius et al. Travel time estimation for urban road networks using low frequency probe vehicle data, Transportation Research B, 2013
• Yilun Wang et al. Travel time estimation of a path using sparse trajectories, KDD 2014
• Zheng Wang et al. Learning to estimate the travel time, KDD 2018.
• Yaguang Li et al. Multi-task Representation Learning for Travel Time Estimation, KDD 2018.
Map Service II
Traffic Estimation and Forecasting
Introduction

• Traffic congesting is wasteful of time, money and energy
  • Traffic congestion costs Americans $124 billion+ direct/indirect loss in 2013.
• Accurate traffic forecasting could substantially improve route planning and mitigate traffic congestion.
Data Source

- Traffic estimation and forecasting
  - Data source: loop detector, GPS trajectory and camera
  - Target: traffic speed in real time and for the future
Traffic Estimation

- Calculate the ground-truth traffic data: multiple source data fusion, spatial-temporal spline.
Traffic Prediction

- **Input:** road network and past $T'$ traffic speed observed at sensors
- **Output:** traffic speed for the next $T$ steps

### Input: Observations

<table>
<thead>
<tr>
<th>Time</th>
<th>Observations</th>
</tr>
</thead>
<tbody>
<tr>
<td>7:00 AM</td>
<td>...</td>
</tr>
<tr>
<td>8:00 AM</td>
<td></td>
</tr>
</tbody>
</table>

### Output: Predictions

<table>
<thead>
<tr>
<th>Time</th>
<th>Predictions</th>
</tr>
</thead>
<tbody>
<tr>
<td>8:10 AM, 8:20 AM, ..., 9:00 AM</td>
<td>...</td>
</tr>
</tbody>
</table>
Challenges for Traffic Forecasting

- Complex Spatial Dependency
- Non-linear, non-stationary Temporal Dynamic
Challenges for Traffic Forecasting

- Spatial relationship among traffic flow is non-Euclidean and directed.
KNN-based Approaches

• Find similar historical traffic time series
  • Extract various features from traffic time series
  • Define similarity metrics between traffic time series

• Calculate the prediction by aggregating next traffic readings of nearest neighbors.

Time Series Methods

- Seasonal Autoregressive Integrated Moving Average (SARIMA)

\[
\left(1 - \sum_{i=1}^{P} \phi_i L^i\right) (1 - L)^d X_t = \delta + \left(1 + \sum_{i=1}^{q} \theta_i L^i\right) \varepsilon_t.
\]

S-ARIMA
Auto-Regressive Integrated Moving Averages (w/Seasonality)

- Support Vector Regression (SVR)

* Picture from http://www.saedsayad.com/support_vector_machine_reg.htm
Traffic Prediction with Latent Space Model (LSM)

- Model the road network as a graph
- Model correlation between road segment with Latent Space model

Define a model that can generate the traffic condition of road network

* Dingxiong Deng et al, Latent Space Model for Road Networks to Predict Time-Varying Traffic. KDD, 2016
Latent Attributes

- Each vertex has **latent** attributes
  - Vertex $i$ has latent attribute vector $\mathbf{u}_i \in \mathbb{R}^{1 \times k}_+\)

<table>
<thead>
<tr>
<th></th>
<th>highway</th>
<th>arterial</th>
<th>business</th>
<th>resident</th>
<th>intersection</th>
<th>non-inter</th>
</tr>
</thead>
<tbody>
<tr>
<td>$i$</td>
<td>0.9</td>
<td>0.1</td>
<td>0.8</td>
<td>0.2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$j$</td>
<td>0.8</td>
<td>0.1</td>
<td>0.7</td>
<td>0.3</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

**Node attribute matrix** $U^{n \times k}$

*(n is the number of vertices)*
Attribute Interaction

- Interaction matrix $B$ between different attributes

<table>
<thead>
<tr>
<th></th>
<th>highway</th>
<th>arterial</th>
<th>business</th>
<th>resident</th>
<th>intersection</th>
<th>non-inter</th>
</tr>
</thead>
<tbody>
<tr>
<td>highway</td>
<td>40</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>arterial</td>
<td></td>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>business</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>residential</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>intersection</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>non-inter</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Attribute interaction matrix $B \in R^{k \times k}$
Speed from Latent Attribute

- Traffic speed between vertices \( i \) and \( j \) (\( i \rightarrow j \)) is a linear combination of the corresponding traffic patterns.

\[
d(i, j) = u_i \times B \times u_j^T
\]

![Diagram showing the calculation of speed between vertices \( i \) and \( j \).]
Basic Graph Model

Graph matrix: $G \in \mathbb{R}^{N \times N}$

Latent properties: $U \in \mathbb{R}^{n \times k}$ and $B \in \mathbb{R}^{k \times k}$

$$\arg \min_{U \geq 0, B \geq 0} J = \| G - U B U^T \|_F^2$$

Non-negative Graph Matrix Factorization (NMF) [1]

Overview of LSM for Road Network

Latent attributes evolve with different timestamps
LSM for Road Network

Basic Graph Model

Overcome the Sparsity

Temporal Effect

Transition Effect

\[
J = \|G - UBU^T\|_F^2
\]

\[
J = \|Y \odot (G - UBU^T)\|_F^2 + \lambda \text{Tr}(U^T LU)
\]

\[
J = \sum_{i=1}^{t} \|Y_i \odot (G_i - U_iBU_i^T)\|_F^2 + \sum_{i=1}^{t} \lambda \text{Tr}(U_i^T LU_i)
\]

\[
J = \sum_{i=1}^{t} \|Y_i \odot (G_i - U_iBU_i^T)\|_F^2 + \sum_{i=1}^{t} \lambda \text{Tr}(U_i^T LU_i) + \sum_{i=2}^{t} \gamma \|U_i - U_{i-1}A\|_F^2
\]

* Dingxiong Deng et al, Latent Space Model for Road Networks to Predict Time-Varying Traffic. KDD, 2016
Experimental Settings

• Dataset
  • March and April, 2014 sensor data with more than 60 million records
  • Two subgraphs of Los Angeles road network

• Baselines:
  • LSM-Naive [Zhang et al. KDD’12]
  • ARIMA [Williams et al. TRB’98], ARIMA-SP
  • SVR [Wu et al. ITS’04], SVR-SP

• Measurement: MAPE, RMSE
Experimental Results

Latent space models (LSM) outperform other baselines.

Latent space models (LSM) outperform other baselines.
Deep Learning for Traffic Forecasting

- Traffic Prediction using Stacked Autoencoder (SAE) together with logistic regression on top of the network for supervised traffic flow prediction

* Yisheng Lv et al. Traffic flow prediction with big data: A deep learning approach, IEEE TITS, 2015
Deep Learning for Extreme Traffic Prediction

• Main Goal: forecasting traffic for extreme including rush-hour and post-accident

Deep Mixture LSTM

- Deep Mixture LSTM is a mixture of LSTM and Autoencoder
  - LSTM: normal condition traffic
  - Autoencoder: accident specific features (time, severity)
  - Merge: linear regression
Experiments

• Baselines:
  • ARIMA: Auto Regressive Integrated Moving Average
  • Random Walk: constant time series with random noise.
  • Historical Average: weighted average of previous seasons
  • Support Vector Regression(SVR): regression using Support Vector Machine

• Data:
  • Traffic: 2,018 sensors from May 19, 2012 to June 30, 2012
  • Accident: 6,811 incidents spread across 1,650 sensors
Rush-Hour Forecasting

- We observe almost 50% improvement for the peak hour traffic forecast.
Post-traffic Forecasting

- Deep Mixture LSTM is roughly 30% better than the baseline methods

Post-accident traffic forecasting MAPE of different forecasting horizons
Convolution Neural Network for Traffic Forecasting

- Model traffic speeds in different locations as a matrix (image).
- Apply convolution to model the spatiotemporal dependency.

* Xiaolei Ma et al. Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction, Sensors, 2017
Traffic Forecasting with Convolution on Graph

- Model spatial dependency with proposed **diffusion convolution on graph**
- Model temporal dependency with **augmented recurrent neural network**

* Yaguang Li et al, Diffusion Convolutional Recurrent Neural Network: Data-driven Traffic Forecasting. ICLR, 2018
Spatial Dependency in Traffic Prediction

- Spatial dependency among traffic flow is **non-Euclidean** and **directed**

---

**Close in Euclidean space**

- **Similar traffic speed**

\[ \text{dist}_{net}(v_i \rightarrow v_j) \neq \text{dist}_{net}(v_i \rightarrow v_j) \]
Spatial Dependency Modeling

• Model the network of traffic sensors, i.e., loop detectors, as a *directed graph*

  [Graph $\mathcal{G} = (\mathbf{V}, \mathbf{A})$]
  
  - Vertices $\mathbf{V}$: $\circ$ sensors
  - Adjacency matrix $\mathbf{A}$: $\rightarrow$ weight between vertices

\[
A_{ij} = \exp \left( - \frac{\text{dist}_{\text{net}}(v_i, v_j)^2}{\sigma^2} \right) \quad \text{if } \text{dist}_{\text{net}}(v_i, v_j) \leq \kappa
\]

$\text{dist}_{\text{net}}(v_i, v_j)$: road network distance from $v_i$ to $v_j$,  
$\kappa$: threshold to ensure sparsity, $\sigma^2$ variance of all pairwise road network distances
Problem Statement

- **Graph signal:** $X_t \in \mathbb{R}^{|V| \times P}$, observation on $G$ at time $t$
  - $|V|$: number of vertices
  - $P$: feature dimension of each vertex.

- **Problem Statement:** Learn a function $g(\cdot)$ to map $T'$ historical graph signals to future $T$ graph signals.

\[ X_{t-T'+1} \quad \ldots \quad X_t \quad \underbrace{\quad g(\cdot) \quad}_{\text{function}} \quad X_{t+1} \quad \ldots \quad X_{t+T} \]
Spatial Dependency Modeling

- Convolution Neural Networks* (CNN) learn meaningful *spatial patterns*
  - State-of-the-art results on image related tasks

Generalize Convolution to Graph

- Diffusion convolution filter: combination of diffusion processes with different steps on the graph.

\[ X_{:,p} \ast_{\mathcal{G}} f_{\theta} = \sum_{k=0}^{K-1} \left( \theta_k (D^{-1}_o A)^k \right) X_{:,p} \]

Transition matrices of the diffusion process

Learning complexity: \( O(K) \)

Example diffusion filter

Centered at

\( \ast_{\mathcal{G}} \): diffusion convolution, \( D_o \): diagonal out-degree matrix.
Generalize Convolution to Graph

- Diffusion convolution filter: combination of **diffusion processes** with different steps on the graph.

\[ X_{:,p} \ast_g f_{\theta} = \sum_{k=0}^{K-1} \left( \theta_{k,1} (D_0^{-1}A)^k + \theta_{k,2} (D_I^{-1}A^\top)^k \right) X_{:,p} \]

Example diffusion filter

Centered at \( \text{Diffusion} \)

• Dual directional diffusion to model upstream and downstream separately

\[ \ast_g : \text{diffusion convolution, } D_0: \text{diagonal out-degree matrix, } D_I: \text{diagonal in-degree matrix} \]
Advantage of Diffusion Convolution

\[ X_{:,p} \ast_{g} f_{\theta} = \sum_{k=0}^{K-1} \left( \theta_{k,1} (D_0^{-1} A)^k + \theta_{k,2} (D_I^{-1} A^\top)^k \right) X_{:,p} \]

- Efficient
  - Learning complexity: \( O(K) \)
  - Time complexity: \( O(K |E|), |E| \) number of edges

- Expressive
  - Many popular convolution operations, including the ChebNet [Defferrard et al., NIPS ’16], can be seen as special cases of the diffusion convolution [Li et al. ICLR ’18].

\( \ast_{g} \): diffusion convolution, \( D_0 \): diagonal out-degree matrix, \( D_I \): diagonal in-degree matrix

* Defferrard, M et al, Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering, NIPS, 2016
* Yaguang Li et al. Diffusion Convolutional Recurrent Neural Network: Data-driven Traffic Forecasting, ICLR, 2018
Model Temporal Dynamics using Recurrent Neural Network

Multi-step ahead prediction with RNN

Current Time

Error Propagation
Teach the model to deal with its own error.

Previous model output is fed into the network

Model prediction
Observation or ground truth
Improve Multi-step ahead Forecasting

- Traffic prediction as a **sequence to sequence** learning problem
  - Encoder-decoder framework

\[ x_1, x_2, x_3 \rightarrow x_4 \]
\[ x_1, x_2, x_3 \rightarrow x_4, x_5, x_6 \]

* Sutskever et al. Sequence to sequence learning with neural networks, NIPS 2014

Backprop errors from multiple steps.

Model prediction
Observation or ground truth

Ground truth becomes unavailable in testing.
Improve multi-step ahead forecasting with **scheduled sampling**

* Bengio, Samy et al. Scheduled sampling for sequence prediction with recurrent neural networks. NIPS 2015
Diffusion Convolutional Recurrent Neural Network

- Diffusion Convolutional Recurrent Neural Network (DCRNN)
  - Model spatial dependency with **diffusion convolution**
  - Sequence to sequence learning with **encoder-decoder** framework
  - Improve multi-step ahead forecasting with **scheduled sampling**

*Yaguang Li et al. Diffusion Convolutional Recurrent Neural Network: Data-driven Traffic Forecasting, ICLR 2018*
Experiment - Datasets

**METR-LA:**
- 207 traffic sensors in Los Angeles
- 4 months in 2012
- 6.5M observations

**PEMS-BAY:**
- 345 traffic sensors in Bay Area
- 6 months in 2017
- 17M observations
Experiments

• Baselines
  • Historical Average (HA)
  • Autoregressive Integrated Moving Average (ARIMA)
  • Support Vector Regression (SVR)
  • Vector Auto-Regression (VAR)
  • Feed forward Neural network (FNN)
  • Fully connected LSTM with Sequence to Sequence framework (FC-LSTM)

• Task
  • Multi-step ahead traffic speed forecasting
Experimental Results

- **DCRNN** achieves the **best performance** for all forecasting horizons for both datasets.
• **w/o temporal**: removing sequence to sequence learning.

• **w/o spatial**: remove the diffusion convolution.

Removing either spatial or temporal modeling results in *significantly worse* results.

![Graph](image)
Related Work

- Traffic Prediction **without** spatial dependency modeling
  - Simulation and queuing theory [Drew 1968]
  - Kalman Filter: [Okutani et al. TRB’83] [Wang et al. TRB’05]
  - ARIMA: [Williams et al. TRB’98] [Pan et al. ICDM’12]
  - Support Vector Regression (SVR): [Muller et al, ICANN' 97] [Wu et al. ITS ‘04]
  - Gaussian process [Xie et al. TRB’10] [Zhou et al. SIGMOD’15]
  - Recurrent neural networks and deep learning: [Lv et al ITS ‘15] [Ma et al. TRC’15] [Yu and Li et al SDM’17]
Related Work

- **Traffic Prediction with spatial dependency modeling**
  - Vector ARIMA  [Williams and Hoel JTE’03], [Chandra et al. ITS’09]
  - Spatiotemporal ARIMA [Kamarianakis et al., TRB’03] [Min and Wynter, TRC’11]
  - k-Nearest Neighbor  [Li et al. ITS’12] [Rice et al. ITS’13]
  - Latent Space Model [Deng et al. KDD’ 16]
  - Convolutional Recurrent Neural Network [Ma et al. Sensors’17]
  - Diffusion Convolutional Recurrent Neural Network [Li et al. ICLR’18]
• [Chandra et al. ITS’09] Chandra, S.R. and Al-Deek, H., Predictions of freeway traffic speeds and volumes using vector autoregressive models. ITS, 2009
• [Li et al. ITS’12] Li, S., Shen, Z. and Xiong, G., A k-nearest neighbor locally weighted regression method for short-term traffic flow forecasting, ITS, 2012
• [Li et al. ICLR’18] Li, Y., Yu, R., Shahabi C., Liu, Y., Diffusion Convolutional Recurrent Neural Network: Data-driven Traffic Forecasting. ICLR, 2018
• [Pan et al. ICDM’12] Pan, B., Demiryurek, U. and Shahabi, C., Utilizing real-world transportation data for accurate traffic prediction. ICDM, 2012
• [Xie et al. TRB’10] Xie, Y., Zhao, K., Sun, Y. and Chen, D., Gaussian processes for short-term traffic volume forecasting. Transportation Research Record, 2010
Decision Service
Intelligent Order-Dispatching System

Order and Driver Forecasting

A Large-Scale Distributed Computing

Platform Efficiency and Customer Experience Optimization
Map Service

Route Planning
The Core of Dispatching

- To minimize cost
- To maximize driver efficiency
- To optimize transportation efficiency

ETA
(Estimated Time of Arrival)

- To estimate the traveling time and the waiting time of each ride
Order-Dispatching Matrix

Diversified Orders

- DiDi Premier
- DiDi Express
- DiDi Taxi

Orders for Premier   Orders for Express   Orders for Taxi

Transport Capacity
Bipartite Graph Matching

Hungarian matching algorithm (also called the Kuhn-Munkres algorithm)
MDP Formulation for Dispatching
Reinforcement Learning

Dispatching
• Every dispatching decision affects future supply (drivers) distribution
• Maximize drivers’ collective income through optimized dispatching, while ensuring good customer experience

Reinforcement Learning
• Focus on long-term reward (e.g. one day)
• Consider future impact of current decision
• Value function is a key quantity to learn
Combining Reinforcement Learning and Combinatorial Optimization

Matching Value

Instant Reward + Future State-Value

Value Functions

Online Planning Step

Offline Learning Step

Historical data

Xu et al., KDD 2018
MDP Definition

Motivation

- Each order dispatch accords to a decision made by the platform
- Formulate order dispatch into a Markov Decision Process (MDP)
- Optimal policy generates the optimal revenue of the platform – by satisfying more requests and maximizing driver’s income

- $\max V_\pi(s) = E_\pi[R_{t+1} + \gamma R_{t+2} + \cdots | S_t = s]$
Learning – Policy Evaluation

Dynamic Programming

**Vacant**: 
\[ V_\pi(S_0) \leftarrow V_\pi(S_0) + \alpha(0 + \gamma V_\pi(S_1) - V_\pi(S_0)) \]

**Serving**: 
\[ V_\pi(S_1) \leftarrow V_\pi(S_1) + \alpha(R + \gamma^2 V_\pi(S_2) - V_\pi(S_1)) \]
Planning – Advantage Function

Link weight:

\[ Q(s, a) \rightarrow A(s, a) \]

\[ = R. + V(s') - V(s) \]

Order’s Value function of driver’s expected finishing state
Value function of driver’s current state
Deep RL for Dispatching
Deep Reinforcement Learning

more adaptive to real-time supply-demand context changes

facilitates learning from multiple cities and times

weights sharing among inputs: location, time, destination, context - better generalization

Wang et al., ICDM 2018
Deep Q-network with action search

- Use historical trips data as training transitions.
- Each trip \( x \) defines a transition of the agent’s state \((s_0, a, r, s_1)\):
  - Current state \( s_0 := (l_0, t_0, f_0) \), location, time, and contextual features
  - Action: the assigned trip;
  - Next-state \( s_1 := (l_1, t_1, f_1) \)
  - Reward: the total fee collected for the trip.
- Implicitly consider pick-up time, trip duration relative to reward

- Training data
- Action search
- Expanded action search
Deep Q-network with action search

- Construct an approximate feasible space for the actions for computing $\max_{a' \in A} Q(s_1, a')$ for the targets.
- Instead of searching through all valid actions, we search within the historical trips originating from the vicinity of $s$:

$$\tilde{A}(s) := \{x_{s_1} | x \in X, B(x_{s_0}) = B(s)\}$$

- The same search procedure is used for evaluation, where we simulate the driver’s trajectory during the day using historical trips data.

- Training data
- Action search
- Expanded action search
Deep Q-network with action search

- Training data
- Action search
- Expanded action search

- Due to training data sparsity in certain spatio-temporal regions, we perform an expanded action search in both spatial and temporal spaces.
Training for multiple cities

Dispatching system supports a large number of cities

- Computationally intensive
- Diverse supply-demand settings

Knowledge transfer

- Leverage common properties: e.g. rush-hour traffic pattern
- Improve learning efficiency

Wang et al., ICDM 2018
Transfer Learning

Idea

- Re-use weights
- Better initial solution

Existing Methods

- Fine-tuning
- Progressive: lateral connections between source city nodes and target city nodes
Correlated Feature Progressive Transfer (CFPT)

Spatio-temporal displacement
- Involve relative transition pattern like distance and time cost

General online features
- Number of idle drivers
- Number of orders created
- Average pick-up time for passengers
- ...
Experiments

Training data: one month of Express Car trip data
Single-driver test environment

DQN v.s. policy evaluation
  - Policy evaluation: max-Q -> mean-Q in mini-batch updates

Transfer learning v.s. no-prior DQN
Spatial transfer
  - Source city to target city
Temporal transfer
  - Same city: one time period to another

<table>
<thead>
<tr>
<th>City</th>
<th>Size</th>
<th>Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Large</td>
<td>Northern</td>
</tr>
<tr>
<td>B</td>
<td>Small</td>
<td>Southern</td>
</tr>
<tr>
<td>C</td>
<td>Medium</td>
<td>Southern</td>
</tr>
<tr>
<td>D</td>
<td>Large</td>
<td>Western</td>
</tr>
</tbody>
</table>
Results

DQN v.s. policy evaluation

- Optimization helps
- Smaller cities with simpler pattern and fewer transitions have smaller advantages
Results

No transfer v.s. transfer

Spatial transfer
- Robustness
- Better initial solution
- Higher converged cumulative rewards
Results

Temporal transfer

(a) City B  (b) City C  (c) City D
Dynamic Ride Sharing
Dynamic Ride Sharing

• Limitations
  • target on single objective
  • rely on inefficient insertion, $O(n^2)$ or $O(n^3)$

• Contributions
  • a unified cost function, generalize three main objectives
  • dynamic programming based insertion, linear time

* Yongxin Tong et al. A Unified Approach to Route Planning for Shared Mobility, VLDB, 2018.
Unified Problem Definition

• Given a set of drivers, a set of requests dynamically arrived, the platform aims to plan route of each driver for serving the requests to minimize the unified cost.

\[ UC(W, R) = \alpha \cdot \sum_{w \in W} D(S_w) + \sum_{r \in R^-} p_r \]

• \( \alpha \): weight, \( p_r \): penalty for rejecting the request

  \[ \begin{cases} \alpha = 1 \\ p_r = \infty \end{cases} \implies \]

  \[ \begin{cases} \alpha = 0 \\ p_r = 1 \end{cases} \implies \]

  \[ \begin{cases} \alpha = \text{unit payment} \\ p_r = \text{fare of the request} \end{cases} \implies \]

* Yongxin Tong et al. A Unified Approach to Route Planning for Shared Mobility, VLDB, 2018.
Core Operation in Ridesharing: Insertion

- Given a worker with current route, a new request is inserted into current route with minimal increased distance, while orders of old requests remain the same.

* Yongxin Tong et al. A Unified Approach to Route Planning for Shared Mobility, VLDB, 2018.
Dynamic Programming based Insertion

- Even though there are $O(n^2)$ pairs, insertion can be implemented by dynamic programming in linear time.

$O(n^2)$ pairs

![Diagram showing the old route and new request]

Dynamic Programming

$$
\text{Dio}[j] = \begin{cases} 
\infty, & \text{if } \text{picked}[j-1] > K_w - K_r \\
\text{Dio}[j-1], & \text{if } \text{det}(l_{j-1}, o_r, l_j) > \text{slack}[j-1] \\
\min\{\text{Dio}[j-1], \text{det}(l_{j-1}, o_r, l_j)\}, & \text{otherwise}
\end{cases}
$$

$$
\text{Plc}[j] = \begin{cases} 
\text{NIL}, & \text{if } \text{picked}[j-1] > K_w - K_r \\
\text{Plc}[j-1], & \text{if } \text{det}(l_{j-1}, o_r, l_j) > \text{slack}[j-1] \\
\text{Plc}[j-1], & \text{if } \text{Dio}[j-1] < \text{det}(l_{j-1}, o_r, l_j) \\
j - 1, & \text{if } \text{Dio}[j-1] \geq \text{det}(l_{j-1}, o_r, l_j)
\end{cases}
$$

* Yongxin Tong et al. A Unified Approach to Route Planning for Shared Mobility, VLDB, 2018.
Supply and Demand Forecasting
Supply and Demand Forecasting

Tong et al., KDD 2017
Supply and Demand Forecasting

DeepST or ST-ResNet

• Spatial proximity: extract features from spatially proximate neighborhoods

• Temporal feature: temporal proximity and periodicity (n-weeks ago, n-days ago, n-hours ago)

Zhang, et al, AAAI 2017
Supply and Demand Forecasting

Deep Multi-View ST Network

- Spatial proximity: extract features from spatially proximate neighborhoods
- Temporal feature: Recurrent Neural network
- Semantic similarity: build semantic graph followed by embedding

Yao, et al, AAAI 2018
Supply and demand forecast and transport capacity dispatch

Supply and demand forecast: predict the future supply and demand conditions possibly happening in a certain region ahead of time.

Driver dispatching: before the occurrence of travel demand, idle drivers are dispatched to the designated region, so as to avoid the great insufficiency of traffic capacity in the target region in the near future.

How to forecast the supply and demand, implementing the intelligent driver dispatching system?

Build the advanced machine learning model based on multi-dimensional feature engineering to forecast the future passenger and driver distributions in any given region of the city.

Feature engineering:
- weather
- traffic condition
- historical observations
- ...

Notations:
- Districtable area
- Order-broadcasting area
- Traffic capacity shortage area

According to the supply and demand forecast, we dispatch the driver from the dispatchable area to the traffic capacity shortage area.

What is the social significance of driver dispatching?

- Resolve the imbalance situation of supply and demand among regions
- Arrange the traffic capacity in advance
- Increase the travel success rate
- Improve the passenger travel experience
- Raise the drivers' income
Smart Transportation
Data Collection, State Estimation, and Prediction

Time-space diagram

2018-4-18

<table>
<thead>
<tr>
<th>Time</th>
<th>MsgType</th>
<th>Msg</th>
</tr>
</thead>
<tbody>
<tr>
<td>16:47:39</td>
<td>Traffic Sign</td>
<td>No Parking</td>
</tr>
<tr>
<td>16:47:39</td>
<td>Traffic Sign</td>
<td>Speed Limit 60km/h</td>
</tr>
<tr>
<td>16:47:39</td>
<td>Traffic Sign</td>
<td>Bus Lane</td>
</tr>
<tr>
<td>16:47:51</td>
<td>Traffic Sign</td>
<td>No Left Turn</td>
</tr>
<tr>
<td>16:48:02</td>
<td>Congestion</td>
<td>Stop in Queue</td>
</tr>
<tr>
<td>......</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Performance Evaluation and Diagnosis

Every movement at the intersection
Real Time Traffic Control

Real-time data processing with distributed computing

Data fusion-mobile sensing + fixed location detection

Traffic states reconstruction

Integrated ramp metering and traffic signal control

Machine learning based controller
Integrated Solution - Smart Transportation Brain

Data Center

Analysis Center

Smart Transportation Brain

Control Center

Lane Control  Smart Dispatch  Path Guidance  Signal Timing  Network Optimization

Government Data

Industry Data

Crowdsourced Data

DiDi Data

Data Fusion

Cloud Computation

AI

Measure System

Simulation
Improving traffic conditions in over 20 cities

Wuhan
Smart Traffic Lights: 170+
Variable Message Signs: 11

Chengdu
Smart Traffic Lights: 120+

Shenzhen
Smart Traffic Lights: 10
Variable Message Signs: 2

Jinan
Smart Traffic Lights: 340+
Variable Message Signs: 82

Nanjing
Smart Traffic Lights: 18

Suzhou
Smart Traffic Lights: 160+

Guangzhou
Smart Traffic Lights: 110+
Electric Vehicle
Motivations

Less Emission Brighter Future!

>1.7 million new energy vehicles in China (the world's largest)

- 400,000 electric cars operating on DiDi’s platform

Supporting the new energy vehicles

- Charging station
- Battery managing system (BMS)
- Dispatch strategy
Charging Station

- Rich Trace Data
- Charging Station Locations
- Station Capacities
- Interactive Map

Spacial Clustering

Demand Prediction

Product

Potential Charging Stations

Supply-demand

Piles needed
BMS (Consumption Prediction)

**Prediction**

**Difference Model**

**Unified Scale**

**Vehicle information**
1. Power output
2. Weight
3. ...

**Weather**
1. Sunny/rainy
2. Temperature
3. ...

**Driving Style**
1. Acc/break freq
2. Max speed
3. ...

**Road Condition**
1. Even?
2. Up/down hill
3. ...

- MAPE distribution
- Counts
- A sample trace fitting
  - truth
  - predicted
Dispatching Strategy

Adapting the unique characteristic of new energy vehicles

I have to charge now
I am 100% full
I have 30% battery

DiDi Brain

I am going 50km away
I am going to a place near charging station
My destination is 10km away
AI for Social Good
AI for Social Good

- Intelligent Medical Hardware
- Barrier-Free
- Reinforcement Learning
- Healthcare
- Disease Warning
- Urban Air Quality Monitoring
- Neural Networks
- Clustering
- Deep Learning
- Environmental Sustainability
- EV Charging Piles Location
- Call for Participant
- Machine Learning
Part 3: Data and Tools for Transportation AI
GAIA Open Dataset
The GAIA Initiative aims to establish strong links between DiDi and academia, and facilitate data-driven research in transportation. It provides a platform for the academia to access anonymized data from real-life urban scenarios.
## Big Data

<table>
<thead>
<tr>
<th>Daily Rides</th>
<th>Data Processed</th>
<th>Routing Requests</th>
<th>New Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>30 Million</td>
<td>4875TB+</td>
<td>40 Billion</td>
<td>100TB+</td>
</tr>
</tbody>
</table>

Every Day
Sharing Anonymized Data

Vehicle Trajectory Data
From: Chengdu and Xi’an, China

- Real
- High Accuracy
- Full Sample
Register
Visit the GAIA Initiative website: GAIA.didichuxing.com/en and click “Apply Now” to register.

Verification & Review
In the Data Center, click “Data Access Application”. Read the user agreement carefully and fill out the application form with your info, then submit the form for verification.

Access Data
Once your application is approved, DiDi will send you an email with instructions for data access. Please check your email.
Outreach

DiDi

Academia

Scenarios

Data

To Redefine the Future of Mobility

GAIA Open Dataset

GAIA.didichuxing.com/en