Data-Driven Mesoscopic Simulation of Large-Scale Surface Transit Networks

MASc Candidate: Bo Wen Wen
Supervisor: Prof. Amer Shalaby
Presentation Outline

- Introduction
- Modelling Framework
- Data
- Model Estimation
- Data-driven Simulation
- Case Study
Introduction
The Nexus Platform

- **Simulation platform**: currently in development, motivated by the need for a high-fidelity multimodal transit network modelling system with capability to:
  - Represents the dynamic behaviour of transit lines and stations
  - Predicts passenger travel behaviour under normal and irregular conditions

- **Scenario analyses**: disruptions, response strategies, and long range planning

1. Srikukenthiran, 2015
The Nexus Platform

- Connects specialized simulators of train operation, pedestrian simulation and surface vehicle movement into a network allowing for modular, multi-modal simulation
Research Motivation

- An accurate and efficient surface transit simulator to connect with Nexus

- Simulation of large-scale transit networks where detailed microsimulation is not needed

- Rapid construction of the transit simulation model with little manual effort
Research Motivation

- Traditional models:
  - Difficult to calibrate and computationally intensive
  - Updated infrequently (out of date)

- Open transit data (AVL, APC, GTFS, AFC, etc.) provides:
  - The potential to capture real world stochasticity
  - Rapidly build models using appropriate methodological tools for big data

- Instead of modelling the kinematics of vehicles, transit vehicle arrivals can be modelled using historical data.
Research Objectives

- Develop segment-based (stop-to-stop) transit simulation model based on running speeds and dwell times

- Measures of effectiveness:
  - Accurate network representations
  - Rapid model construction
  - Efficient simulation
Modelling Framework
Modelling Framework
Modelling Framework

- Arrival Time
- Dwell Time
- Departure Time
- Running Speed
- Arrival Time
Modelling Framework
Model Type 1: Basic Analysis

- Route level model

- This type of model accounts for:
  - temporal effects:
    - time of day, and day of the week
  - transit operational characteristics:
    - headway, delay, and previous speeds.
  - basic link characteristics:
    - link distance, link name
Model Type 2: Advanced Analysis

- Network level model

- This type of model accounts for:
  - temporal effects
  - transit operational characteristics
  - expanded link characteristics:
    - stop locations, link distances, link name (link identification), number of signalized intersections, left and right turns made by transit vehicles between stops, traffic and pedestrian volumes
  - route characteristics
    - dedicated right of way, streetcar versus bus route, disruptions, road restrictions or incidents, precipitation.
Data Requirements

- **Model 1: Basic Analysis**
  - AVL or GPS traces of transit vehicle trips
  - Schedule information about the route

- **Model 2: Advanced Analysis**
  - AVL data streams for the entire network
  - GTFS transit network schedules
  - Signalized intersection locations
  - Intersection volume data
  - Road restriction data streams
  - Weather data streams
Data
Methods - Automatic Data Collection

- Manual download procedure for archival data
  - Manually Download Archival Data
  - Unpack archive and validate data
  - Convert archive to XML Object
  - Save XML File

- Automatic download procedure for real-time online API Data
  - Check Current Time
    - If greater than end of download time, **save and terminate**.
    - If less, **schedule next download request**.
  - Sends Asynchronous WebClient Download Request
    - Wait for response
    - Get Web API response object
  - Data Conversion
    - Convert to defined structure
    - Eliminate duplicates
  - Save
    - If reached schedule save, add to memory object and save to disk.
    - Otherwise, save to memory object.
Methods - Automatic Data Collection

- For archival data, retrieval can be performed periodically
  - **General Transit Feed Specification (GTFS):**
    - [Open Data Toronto GTFS data archive](#)
  - **Signalized intersection locations and volume:**
    - [Open Data Toronto active archive](#)

- Periodically sends web requests to retrieve real-time data from public APIs throughout the data collection period
  - **Automatic Vehicle Location (AVL):**
    - [Nextbus real-time data streams](#), 20 seconds resolution
  - **Road restriction:**
    - [Open Data Toronto real-time data streams](#), periodic updates
  - **Weather:**
    - [OpenWeatherMap real-time data streams](#), 3-hour precipitation
Methods - Data Processing

- Program procedure for data processing

- Processes unstructured location and feature data into structured and defined variables

- Preprocesses the data to
  - exclude duplicate points and
  - invalid points (illogical locations)
Methods - Data Processing

- Use AVL and GTFS data to compute various transit operational characteristics.
  - Trip construction
  - Trip matching based on trip geometry
  - Compute trip characteristics:
    - Arrival times,
    - Dwell times,
    - Headway,
    - Delay, etc.

- Spatially and temporally matched additional data to transit trips
  - Signalized intersection location and volume
  - Road restrictions
  - Weather
## Methods – Variable Definition

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Description</th>
<th>Variable Type</th>
<th>Typ. Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>RunningSpeed</td>
<td>Arrival to arrival speed between two stops, dependent var. for running speed model</td>
<td>Continuous</td>
<td>0 to 120 kph</td>
</tr>
<tr>
<td>DwellingTime</td>
<td>Dwell time at the start stop, dependent variable for dwell time model</td>
<td>Continuous</td>
<td>0 to 300 secs</td>
</tr>
<tr>
<td>RouteCode.f</td>
<td>Route Code of travelling vehicle</td>
<td>Categorical</td>
<td>163 levels</td>
</tr>
<tr>
<td>hasIncident.f</td>
<td>If the link segment has road restriction</td>
<td>Categorical</td>
<td>0, 1</td>
</tr>
<tr>
<td>prevLinkRunningSpeed</td>
<td>Previous Running Speed upstream of the current link</td>
<td>Continuous</td>
<td>0 to 120 kph</td>
</tr>
<tr>
<td>prevTripRunningSpeed</td>
<td>Previous Trip’s Running Speed on the current link</td>
<td>Continuous</td>
<td>0 to 120 kph</td>
</tr>
<tr>
<td>Day.f</td>
<td>Day of week</td>
<td>Categorical</td>
<td>0 to 6</td>
</tr>
<tr>
<td>Time_mins</td>
<td>Time of day in minutes since start of study period</td>
<td>Continuous</td>
<td>0 to 86,400 mins</td>
</tr>
<tr>
<td>linkDist</td>
<td>Distance of the current link</td>
<td>Continuous</td>
<td>0 to 11,600 m</td>
</tr>
<tr>
<td>Delay</td>
<td>Estimated schedule delay experienced by the vehicle on the link</td>
<td>Continuous</td>
<td>-1000 to 5000 s</td>
</tr>
<tr>
<td>Headway Ratio</td>
<td>The Ratio between Scheduled and Estimated headway of the vehicle at a stop</td>
<td>Continuous</td>
<td>0 to 30</td>
</tr>
<tr>
<td>totalPptn</td>
<td>Total precipitation reported at the nearest weather station to current link</td>
<td>Continuous</td>
<td>0 to 10 mm</td>
</tr>
<tr>
<td>num_VehLtTurns</td>
<td>Number of Left Turns by the transit vehicle on the link</td>
<td>Categorical</td>
<td>0 to 2</td>
</tr>
<tr>
<td>num_VehRtTurns</td>
<td>Number of Right Turns by the transit vehicle on the link</td>
<td>Categorical</td>
<td>0 to 3</td>
</tr>
<tr>
<td>num_VehThroughs</td>
<td>Number of through movements at intersections made by the transit vehicle</td>
<td>Categorical</td>
<td>0 to 14</td>
</tr>
<tr>
<td>num_TSP_equipped</td>
<td>Number of TSP equipped intersections on the link</td>
<td>Categorical</td>
<td>0 to 6</td>
</tr>
<tr>
<td>num_PedCross</td>
<td>Number of pedestrian crossings on the link</td>
<td>Categorical</td>
<td>0 to 3</td>
</tr>
<tr>
<td>sum_SigIntxnApproach</td>
<td>Total number of signalized approaches of the intersections on the link</td>
<td>Categorical</td>
<td>0 to 49</td>
</tr>
<tr>
<td>avgVehVol</td>
<td>Average vehicle volume of the link</td>
<td>Categorical</td>
<td>0 to 20,000</td>
</tr>
<tr>
<td>avgPedVol</td>
<td>Average pedestrian volume of the link</td>
<td>Categorical</td>
<td>0 to 10,000</td>
</tr>
<tr>
<td>isStartStopNearSided.f</td>
<td>If start stop is near sided</td>
<td>Categorical</td>
<td>0 or 1</td>
</tr>
<tr>
<td>isEndStopFarSided.f</td>
<td>If end stop is far sided</td>
<td>Categorical</td>
<td>0 or 1</td>
</tr>
<tr>
<td>isStreetcar.f</td>
<td>If the route on the link a streetcar route</td>
<td>Categorical</td>
<td>0 or 1</td>
</tr>
<tr>
<td>isSeparatedROW.f</td>
<td>If the link on the route separated right-of-way</td>
<td>Categorical</td>
<td>0 or 1</td>
</tr>
<tr>
<td>linkName</td>
<td>The name of the link</td>
<td>Categorical</td>
<td>9267 levels</td>
</tr>
</tbody>
</table>

*Italicized variables are used in network level models
Model Estimation
Methods – Model Estimation

- **Running Speed Regression Models**
  - Multiple Linear Regression (MLR)
  - Support Vector Machine (SVM)
  - Linear Mixed Effect Model (LME)
  - Regression Tree (RT)
  - Random Forest (RF)

- **Dwell Time Model**
  - dwell times at transit stops followed the lognormal distribution $^{1-5}$

Running Speed Model Estimation

- Program procedure for estimating regression models

- Running Speed Model trained in R, using R.Net via C#
  - Efficient data manipulation (with R data.table)
  - Open source machine learning packages
  - Rapid model prototyping
Multiple Linear Regression (MLR)

- Based on ordinary least squares.
- Four fundamental assumptions:
  - Linear relationships
  - Homoscedasticity
  - Normally distributed errors
  - Independency

- The general form of Multiple Linear Regression model:

\[ Y = a + b_1X_1 + b_2X_2 + \ldots + b_iX_i + \varepsilon \]

  - \( Y \): response variable,
  - \( b_i \): estimated coefficients for predictor variables,
  - \( X_i \): predictor variables,
  - \( \varepsilon \): residuals

1. Marill, 2004
Support Vector Machine (SVM)

- Based on hyperplane margin optimization
  - Edge training points “supports” the minimum margin vector

- Kernel Functions
  - Linear: fast
  - Polynomial: can become too wavy, and it is very slow.
  - Radial Basis function: commonly used, most flexible, but slower than linear kernel.

- Different loss functions determines how model is trained:
  - $\nu$-SVR: controls number of vectors
  - $\epsilon$-SVR: penalizes errors

- $\epsilon$-SVR is most suitable
  - Consistent objective in reducing errors
  - Need to address overfitting with cross-validation

1. Chang and Lin, 2011;
2. Scikit-learn developers, 2014
Linear Mixed Effect Model (LME)

- Accounts for the random sampling variations due to repeated measurements. ¹
  - Deals with heteroscedasticity

- Models random effects by:
  - Varying intercepts
  - Varying slopes

- Same assumptions for each level of the random effects as MLR:
  - Linear relationships
  - Normally distributed errors

¹ Bates et al., 2015;
² Daniel Von, 2014;
³ Human Language Processing (HLP) lab at the University of Rochester, 2014
Regression Tree (RT) ¹

- Partition to determine data clusters.
- Construction of trees are based on splitting criteria.
- Aims to minimize Gini impurity, thus reduce probability of misclassifications.
- Variables that affects the split the most is the most important.

- Complexity and depth of tree are determined by
  - complexity parameter (cp)
  - minimum split criteria
  - prune cp
- Tree pruning with cross-validation can minimize overfitting.

1. Terry M. Therneau and Elizabeth J. Atkinson, 2017;
2. Charpentier, 2013
Random Forest (RF)

- Grow a number of trees based on random draws of the original samples (with replacements) \(^1\)
- An ensemble method:
  - Each tree is a weak learner, but collectively are strong
  - The result from all the trees produces a single prediction
- Works well for clustered data and can replicate complex relationships
- Each draw is independent
- Low correlation needed between residuals and between trees
- Shown not to overfit and reduce bias

1. Breiman, 2001;
2. R. Hänsch and O. Hellwich, 2015
Comparisons of Running Speed Models

- Model Fitness
  \[ R^2 = 1 - \frac{SS_R/df_e}{SS_T/df_t}, \text{df}_e = n - 1, \text{df}_t = n - p - 1 \]

- Mean absolute percentage error:
  \[ MAPE = \frac{1}{n} \sum_{i=0}^{n} \left| \frac{\hat{y}_i - y_i}{y_i} \right| \times 100\% \]

- Mean absolute error:
  \[ MAE = \frac{1}{n} \sum_{i=0}^{n} |\hat{y}_i - y_i| \]

- Relative absolute error:
  \[ RAE = \frac{\sum_{i=0}^{n} |\hat{y}_i - y_i|}{\sum_{i=0}^{n} |y_i - \bar{y}|} \]
Comparisons of Running Speed Models

- Root mean square error:
  \[ RMSE = \sqrt{\frac{1}{n} \sum_{i=0}^{n} (\hat{y}_i - y_i)^2} \]

- Root relative square error:
  \[ RRSE = \sqrt{\frac{\sum_{i=0}^{n} (\hat{y}_i - y_i)^2}{\sum_{i=0}^{n} (y_i - \bar{y})^2}} \]

- Relative differences in RMSE (RD):
  \[ RD = \frac{\text{RMSE}_i - \text{RMSE}_{MLR}}{\text{RMSE}_{MLR}} \]

- Training Time and Test Prediction Time
Methods – Model Estimation

- Running Speed Regression Models
  - Multiple Linear Regression (MLR)
  - Support Vector Machine (SVM)
  - Linear Mixed Effect Model (LME)
  - Regression Tree (RT)
  - Random Forest (RF)

- Dwell Time Model
  - dwell times at transit stops followed the lognormal distribution

Dwell Time Model Estimation

- Program procedure for estimating distribution models

- Dwell Time Model trained in native C#
  - Stop-based models, trained using historical dwell times at the stop
  - Lognormal distribution
  - Open source statistical package (with Math.NET Numerics)
Dwell Time Models

- Model Estimation: Lognormal Distribution
  - Estimation of log mean parameter
    \[ \hat{\mu} = \frac{\sum_{i=1}^{k} \ln x_k}{n} \]
  - Estimation of shape parameter
    \[ \hat{\sigma}^2 = \frac{\sum_{i=1}^{k} (\ln x_k - \hat{\mu})^2}{n} \]

- Model evaluation: chi-squared goodness of fit
  \[ \chi^2 = \sum (P_i - O_i)^2 / O_i \]

1. Li et al., 2012
Data-driven Simulation
Model Simulation

- Program procedure for simulation

- Base case scenario used transit schedule departures from terminals with no short turns

- Running speed and dwell time models predicted mesoscopic transit movements
Model Simulation – Iterative predictions

- Start Simulator
  - GTFS Schedule Trip Data
    - Trip Initialization
    - Simulated Trip Data (many trips)
  - Processed Test Data
    - Data Cross Reference
    - Simulator Trip Processing Tasks

- Is Trip Ready
  - Yes: Add to Batch, Generate Predictions
  - No: Update Trip Data
    - Update Trip Data
      - No: End and Update Trip Data
      - Yes: Wait for trips in other batches to complete
Results

Case Study: Toronto Transit Commission network
Case Study: the TTC network

- The Toronto Transit Commission (TTC) provides public transit in the city of Toronto.
  - Population of Toronto: 2.8 Million

- 4 subway/rail lines, 11 streetcar routes, and over 140 bus routes

- Period of the case study
  - Training Data: 2017-02-28 to 2017-03-02, 6AM to 9AM (AM Peak)
  - Test Data: 2017-03-07 to 2017-03-09, 6AM to 9AM
Summary of Data during study period

- **GTFS**
  - 8304 Trips (typical weekday AM peak)

- **AVL**
  - 8381 Trips (Feb 28), 8350 Trips (Mar 1), 8403 Trips (Mar 2), 8428 Trips (Mar 7), 8395 Trips (Mar 8), 8414 Trips (Mar 9)

- **Road Restrictions**
  - 734 Events (Feb 28 to Mar 2), 766 Events (Mar 7 to Mar 9)

- **Weather**
  - 72 Records (per day)

- **Traffic intersections**
  - 2269 Records (intersection volumes)
  - 71 Records (minor intersections)
## Running Speed Model Results (Network)

<table>
<thead>
<tr>
<th>Model Type</th>
<th>MLR</th>
<th>SVM</th>
<th>LME</th>
<th>RT</th>
<th>RF (100 trees)</th>
</tr>
</thead>
<tbody>
<tr>
<td>R Package</td>
<td>MASS</td>
<td>liquidSVM</td>
<td>LME4</td>
<td>RPART</td>
<td>RANGER</td>
</tr>
<tr>
<td>$R^2$</td>
<td>0.277</td>
<td>0.265</td>
<td>0.387</td>
<td>0.225</td>
<td>0.359</td>
</tr>
<tr>
<td>MAPE</td>
<td>0.355</td>
<td>0.358</td>
<td>0.311</td>
<td>0.372</td>
<td>0.325</td>
</tr>
<tr>
<td>MAE</td>
<td>7.625</td>
<td>7.677</td>
<td>6.902</td>
<td>7.911</td>
<td>7.109</td>
</tr>
<tr>
<td>RAE</td>
<td>0.831</td>
<td>0.837</td>
<td>0.752</td>
<td>0.862</td>
<td>0.775</td>
</tr>
<tr>
<td>RRSE</td>
<td>0.850</td>
<td>0.858</td>
<td>0.783</td>
<td>0.881</td>
<td>0.800</td>
</tr>
<tr>
<td>Reduction in RMSE</td>
<td>-</td>
<td>-0.9%</td>
<td>7.9%</td>
<td>-3.5%</td>
<td>5.9%</td>
</tr>
<tr>
<td>Training Time (min.)</td>
<td>0.419</td>
<td>36.272</td>
<td>2.629</td>
<td>1.021</td>
<td>14.681</td>
</tr>
<tr>
<td>Prediction Time (min.)</td>
<td>0.036</td>
<td>3.249</td>
<td>0.049</td>
<td>0.015</td>
<td>0.331</td>
</tr>
</tbody>
</table>
# Running Speed Model Results (504-King)

* Route-level model trained using data from 504-King only.

<table>
<thead>
<tr>
<th>Model Type*</th>
<th>MLR</th>
<th>SVM</th>
<th>LME</th>
<th>RT</th>
<th>RF (100 trees)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>R Package</strong></td>
<td>MASS</td>
<td>liquidSVM</td>
<td>LME4</td>
<td>RPART</td>
<td>RANGER</td>
</tr>
<tr>
<td><strong>R²</strong></td>
<td>0.107</td>
<td>0.115</td>
<td>0.223</td>
<td>0.102</td>
<td>0.153</td>
</tr>
<tr>
<td><strong>MAPE</strong></td>
<td>0.329</td>
<td>0.326</td>
<td>0.296</td>
<td>0.330</td>
<td>0.318</td>
</tr>
<tr>
<td><strong>MAE</strong></td>
<td>5.127</td>
<td>5.077</td>
<td>4.726</td>
<td>5.134</td>
<td>4.982</td>
</tr>
<tr>
<td><strong>RAE</strong></td>
<td>0.940</td>
<td>0.931</td>
<td>0.866</td>
<td>0.941</td>
<td>0.913</td>
</tr>
<tr>
<td><strong>RRSE</strong></td>
<td>0.945</td>
<td>0.941</td>
<td>0.882</td>
<td>0.947</td>
<td>0.920</td>
</tr>
<tr>
<td><strong>Reduction in RMSE</strong></td>
<td>-</td>
<td>0.5%</td>
<td>6.7%</td>
<td>-0.3%</td>
<td>2.6%</td>
</tr>
<tr>
<td><strong>Training Time (sec.)</strong></td>
<td>0.017</td>
<td>31.790</td>
<td>0.327</td>
<td>0.662</td>
<td>3.838</td>
</tr>
<tr>
<td><strong>Prediction Time (sec.)</strong></td>
<td>0.011</td>
<td>2.286</td>
<td>0.076</td>
<td>0.012</td>
<td>0.158</td>
</tr>
</tbody>
</table>
Running Speed Model Results

- Sample size
  - Network level: training = 593,234, test = 600,351
  - Route level (504-King only): training = 12,827, test = 12,612

- RT and SVM did not provide improvements over MLR
- SVM provided small improvements over MLR for route level models

- LME model yields the best result with:
  - varying intercept model
  - link identification (link name) as the random effect variable

- RF did well and provided a more flexible implementation
  - allows new links, whereas LME model does not
- LME is more computationally efficient than RF.
Dwell Time Models: Parameters
Dwell Time Model: Observed vs Predicted
Simulation Model Results

- Simulation scenario:
  - Weekday schedule
  - On-time terminal schedule departure, if possible.
  - No short turns
  - Road conditions from test day: 2017-03-08, 6AM to 9AM
    - 704 road restriction events (Mar 8 only)
    - 72 weather records per day
    - Intersection attribute data for links

- Simulations using RF and LME were generated.

- Comparisons of vehicle trajectories with time-distance diagrams.

- Model Validations
  - Route level with route speeds
  - Stop level with stop delays
Simulation Model Results - RF

504 KING EB - Scheduled
504 KING EB - Observed
504 KING EB - Simulated

504 KING WB - Scheduled
504 KING WB - Observed
504 KING WB - Simulated
Simulation Model Results - LME
Model Validations – Route Speeds

Random forest

Linear Mixed Effect
Model Validations – Stop Delays

Random forest

Linear Mixed Effect
Findings

- Running speed model comparisons
  - LME model accuracy outperformed MLR by 8%
  - RF model accuracy outperformed MLR by 6%
  - LME has lower training time, but requires repeated observations from existing links.

- Lognormal dwell time introduce realistic stochasticity into vehicle movements.

- Simulation model prediction runtimes
  - RF (ranger package): 36 minutes
  - LME (lme4 package): 1 minute
Findings

- A data-driven transit simulation model
  - replicated instances of vehicle bunching, distribution of dwell times, and stochastic patterns of delays and headways

- Validation results suggests the need to incorporate:
  - Effect of traffic congestion
  - Signal delays
  - Vehicle short-turns
Future Research

- Model the effects of short-turning vehicles

- Incorporate congestion data

- Advanced dwell time models to incorporate passenger demand
  - Allows reallocation of passenger demand
  - Stop addition, relocation, and removals

- Continuous model training for streaming data
Acknowledgements