Real-time prediction of bus arrival using joint models of vehicle and road states

Tom Elliott

NZSA+ORSNZ 2016

Supervisor: Professor Thomas Lumley

THE UNIVERSITY OF AUCKLAND
Te Whare Wānanga o Tāmaki Makaurau
NEW ZEALAND

SCIENCE
DEPARTMENT OF STATISTICS

tomelliott.co.nz @tomelliottnz

Overview

1. A quick motivation
2. Two real-time models: vehicle (particle filter) & road (Kalman filter)
3. Predicting arrival times

What's wrong with the current† system?

What's wrong with the current† system?

† Specifically Auckland Transport

⇒ applicable to any public transport system using GTFS

What's wrong with the current[†] system?

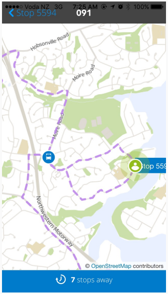
- Prediction inaccuracy

What's wrong with the current† system?

- Prediction inaccuracy
- Prone to errors

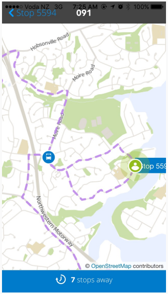
What's wrong with the current† system?

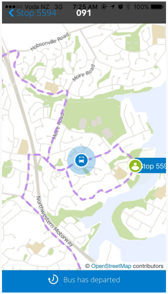
- Prediction inaccuracy
- Prone to errors



What's wrong with the current† system?

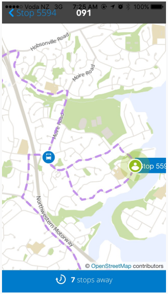
- Prediction inaccuracy
- Prone to errors

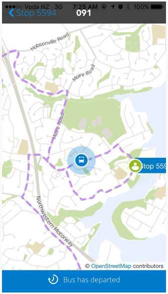


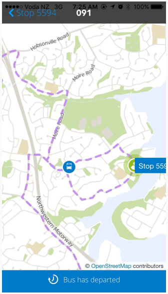


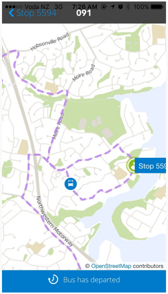
What’s wrong with the current[†] system?

- Prediction inaccuracy
- Prone to errors



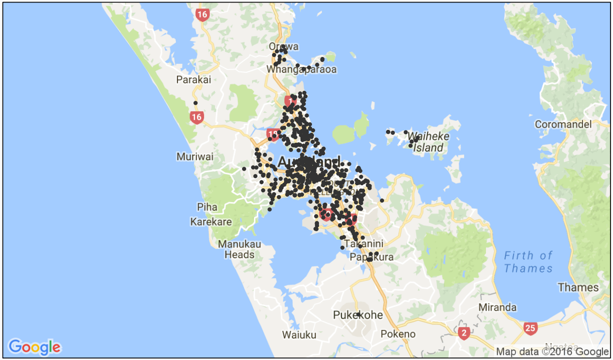






What's wrong with the current† system?

- Prediction inaccuracy
- Prone to errors
- Only specific (small) subsets used — if any!



Vehicle State Model

Vehicle State Model

Goal: use observations of bus location (GPS) ...

$$\mathbf{Y}_k = \begin{bmatrix} \phi_k \\ \lambda_k \\ t_k \end{bmatrix} = \begin{bmatrix} \text{latitude (degrees)} \\ \text{longitude (degrees)} \\ \text{timestamp} \end{bmatrix}$$

Vehicle State Model

Goal: use observations of bus location (GPS) ...

$$\mathbf{Y}_k = \begin{bmatrix} \phi_k \\ \lambda_k \\ t_k \end{bmatrix} = \begin{bmatrix} \text{latitude (degrees)} \\ \text{longitude (degrees)} \\ \text{timestamp} \end{bmatrix}$$

... to infer **unobservable vehicle state** ...

$$\mathbf{X}_k = \begin{bmatrix} d_k \\ v_k \\ s_k \\ \vdots \end{bmatrix} = \begin{bmatrix} \text{distance into trip (meters)} \\ \text{velocity/speed } (ms^{-1}) \\ \text{last visited stop} \\ \vdots \end{bmatrix}$$

Vehicle State Model

Goal: use observations of bus location (GPS) ...

$$\mathbf{Y}_k = \begin{bmatrix} \phi_k \\ \lambda_k \\ t_k \end{bmatrix} = \begin{bmatrix} \text{latitude (degrees)} \\ \text{longitude (degrees)} \\ \text{timestamp} \end{bmatrix}$$

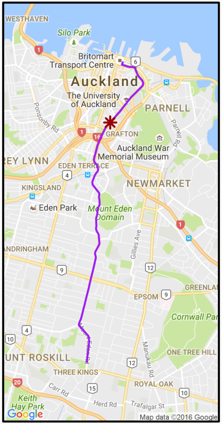
... to infer **unobservable vehicle state** ...

$$\mathbf{X}_k = \begin{bmatrix} d_k \\ v_k \\ s_k \\ \vdots \end{bmatrix} = \begin{bmatrix} \text{distance into trip (meters)} \\ \text{velocity/speed } (ms^{-1}) \\ \text{last visited stop} \\ \vdots \end{bmatrix}$$

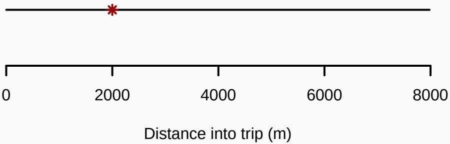
... in real time.

Vehicle State Model

$\mathbf{Y}_k$



$\mathbf{X}_k$ (first component, d_k)



Example: Route 274, Britomart to Three Kings

Vehicle State Model: Particle Filter

- Represent $\mathbf{X}_k$ by a sample of point-estimates (particles) $\mathbf{x}_k^{(i)}$
  1. generate sample of plausible vehicle state predictions
  2. remove predictions no longer plausible, given observation
- Flexible modeling framework, fewer assumptions
- Better coverage of possible states (**multimodality**, robust)
- Intuitive likelihood function

Vehicle State Model: Particle Filter

Step 1: predict

- Start with vehicle state at previous observation, $\mathbf{X}_{k-1} = \{\mathbf{x}_{k-1}^{(i)} : i = 1, \ldots, N\}$

Vehicle State Model: Particle Filter

Step 1: predict

- Start with vehicle state at previous observation, $\mathbf{X}_{k-1} = \{\mathbf{x}_{k-1}^{(i)} : i = 1, \ldots, N\}$
- **Transition** each particle independently

$$\mathbf{x}_k^{(i)} = f\left(\mathbf{x}_{k-1}^{(i)}, \sigma_v^2, \ldots\right)$$

Vehicle State Model: Particle Filter

Step 1: predict

- Start with vehicle state at previous observation, $\mathbf{X}_{k-1} = \{\mathbf{x}_{k-1}^{(i)} : i = 1, \ldots, N\}$
- **Transition** each particle independently

$$\mathbf{x}_k^{(i)} = f(\mathbf{x}_{k-1}^{(i)}, \sigma_v^2, \ldots)$$

1. Add system noise

$$v_k^{(i)} \sim \mathcal{N}_T(v_{k-1}^{(i)}, \sigma_v^2), \qquad 0 \leq v_k^{(i)} \leq \mathbf{v}_{\max}$$

$\mathbf{v}_{\max} \approx$ road speed limit

Vehicle State Model: Particle Filter

Step 1: predict

- Start with vehicle state at previous observation, $\mathbf{X}_{k-1} = \{\mathbf{x}_{k-1}^{(i)} : i = 1, \ldots, N\}$
- **Transition** each particle independently

$$\mathbf{x}_k^{(i)} = f(\mathbf{x}_{k-1}^{(i)}, \sigma_v^2, \ldots)$$

1. Add system noise
2. Move particles along route (Law of Motion)

$$d_k^{(i)} = d_{k-1}^{(i)} + (t_k - t_{k-1}) v_k^{(i)}$$

Vehicle State Model: Particle Filter

Step 1: predict

- Start with vehicle state at previous observation, $\mathbf{X}_{k-1} = \{\mathbf{x}_{k-1}^{(i)} : i = 1, \ldots, N\}$
- **Transition** each particle independently

$$\mathbf{x}_k^{(i)} = f(\mathbf{x}_{k-1}^{(i)}, \sigma_v^2, \ldots)$$

1. Add system noise
2. Move particles along route (Law of Motion)
3. What about intermediate stop(s)?
   - Does the particle stop? $p_{s_k}^{(i)} \sim \mathrm{Bernoulli}(\pi_{s_k})$
   - If so, for how long? $\bar{t}_{s_k} \sim \mathcal{E}(\tau_{s_k})$
   - **Dwell time** $= p_{s_k}^{(i)}(\gamma + \bar{t}_{s_k})$

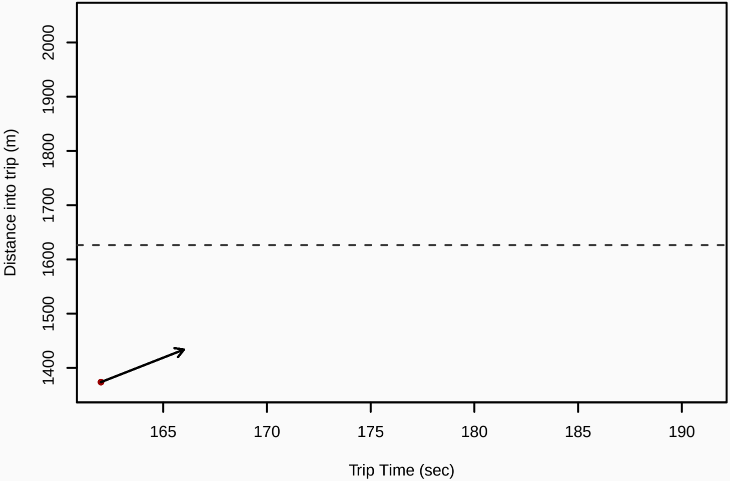
     γ = minimum dwell time (deccelerate/accelerate, open/close doors)

     $\bar{t}_{s_k}$ = passengers on/off

Vehicle State Model: Particle Filter

Step 1: predict

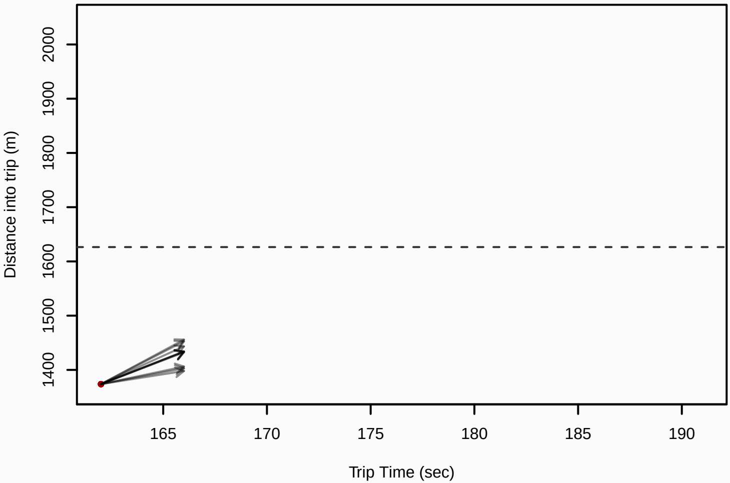
Example ($N = 10$ particles)



Vehicle State Model: Particle Filter

Step 1: predict

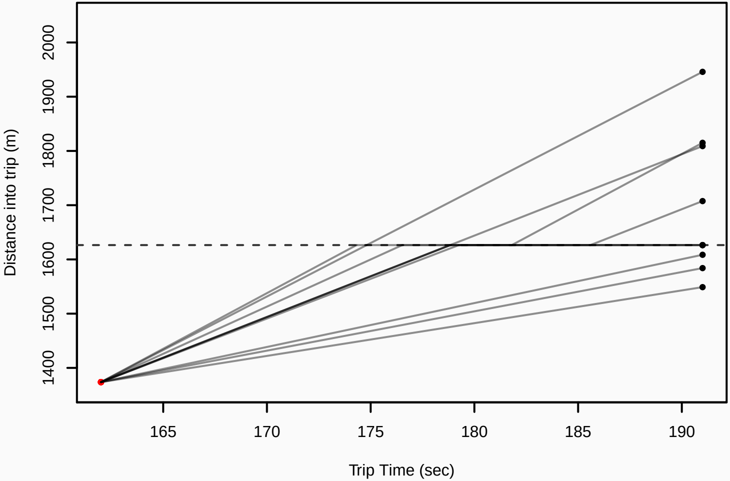
Example ($N = 10$ particles)



Vehicle State Model: Particle Filter

Step 1: predict

Example ($N = 10$ particles)



Vehicle State Model: Particle Filter

Step 2: update

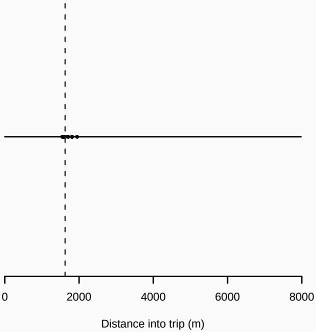
- Likelihood function $\ell(\mathbf{Y}_k|\mathbf{x}_k^{(i)}, \sigma_y^2)$

Vehicle State Model: Particle Filter

Step 2: update

- Likelihood function $\ell(\mathbf{Y}_k | \mathbf{x}_k^{(i)}, \sigma_y^2)$
  - Transform particles onto flat plane

$$\mathbf{z}_k^{(i)} = g(h(\mathbf{x}_k^{(i)}) | \mathbf{Y}_k)$$



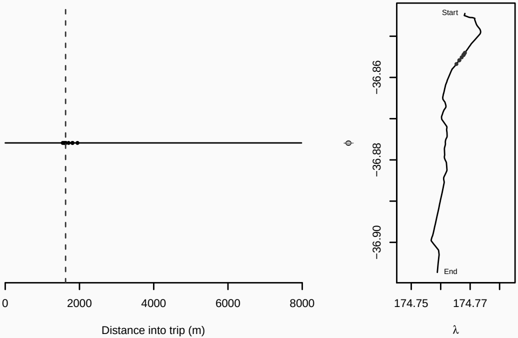
Vehicle State Model: Particle Filter

Step 2: update

- Likelihood function $\ell(\mathbf{Y}_k | \mathbf{x}_k^{(i)}, \sigma_y^2, h)$
  - Transform particles onto flat plane

$$\mathbf{z}_k^{(i)} = g(h(\mathbf{x}_k^{(i)}) | \mathbf{Y}_k)$$

h: measurement function (distance into trip → lat/lon)



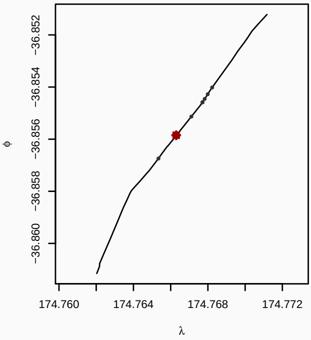
Vehicle State Model: Particle Filter

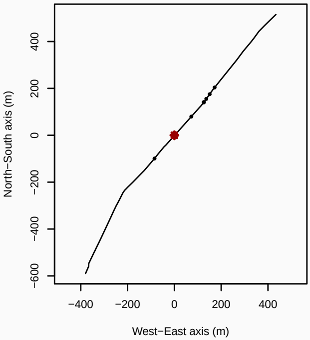
Step 2: update

- Likelihood function $\ell(\mathbf{Y}_k | \mathbf{x}_k^{(i)}, \sigma_y^2, h, g)$
  - Transform particles onto flat plane

$$\mathbf{z}_k^{(i)} = g(h(\mathbf{x}_k^{(i)}) | \mathbf{Y}_k)$$

$g(\cdot | \mathbf{Y}_k)$: projection centered on $\mathbf{Y}_k$, 1 unit = 1 meter in all directions



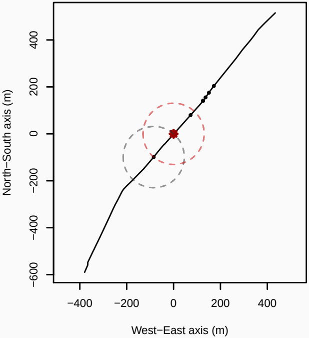


Vehicle State Model: Particle Filter

Step 2: update

- Likelihood function $\ell(\mathbf{Y}_k | \mathbf{x}_k^{(i)}, \sigma_y^2, h, g)$
  - Transform particles onto flat plane
  - Bivariate normal likelihood, $g(\mathbf{Y}_k | \mathbf{Y}_k) = \mathbf{0}$

$$\mathbf{Y}_k | \mathbf{z}_k^{(i)}, \sigma_y^2 \sim \mathbf{z}_k^{(i)} | \mathbf{Y}_k, \sigma_y^2 \sim \mathcal{N}_2(\mathbf{0}, \sigma_y^2 I_2) \qquad (\sigma_y^2 = \text{GPS error})$$



Vehicle State Model: Particle Filter

Step 2: update

- Likelihood function $\ell(\mathbf{Y}_k|\mathbf{x}_k^{(i)}, \sigma_y^2, h, g)$
  - Transform particles onto flat plane
  - Bivariate normal likelihood, $g(\mathbf{Y}_k|\mathbf{Y}_k) = \mathbf{0}$

$$\mathbf{Y}_k|\mathbf{z}_k^{(i)}, \sigma_y^2 \sim \mathbf{z}_k^{(i)}|\mathbf{Y}_k, \sigma_y^2 \sim \mathcal{N}_2(\mathbf{0}, \sigma_y^2 I_2) \qquad (\sigma_y^2 = \text{GPS error})$$

  - For each particle

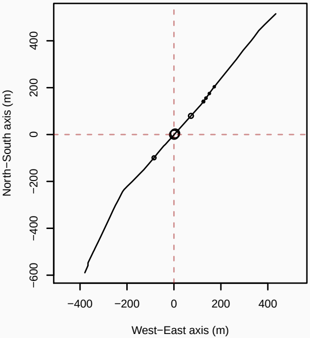
$$\ell(\mathbf{Y}_k|\mathbf{x}_k^{(i)}, \sigma_y^2, h, g) \propto e^{-\frac{1}{2\sigma^2}\left((\mathbf{z}_k^{(i)})^T \mathbf{z}_k^{(i)}\right)}$$

Vehicle State Model: Particle Filter

Step 2: update

- Likelihood function $\ell(\mathbf{Y}_k|\mathbf{x}_k^{(i)}, \sigma_y^2, h, g)$
- Compute weights

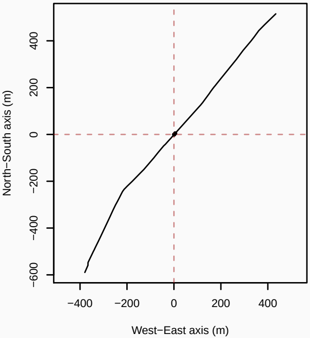
$$w_i = \frac{\ell(\mathbf{Y}_k|\mathbf{x}_k^{(i)})}{\sum_{j=1}^{N} \ell(\mathbf{Y}_k|\mathbf{x}_k^{(j)})}$$



Vehicle State Model: Particle Filter

Step 2: update

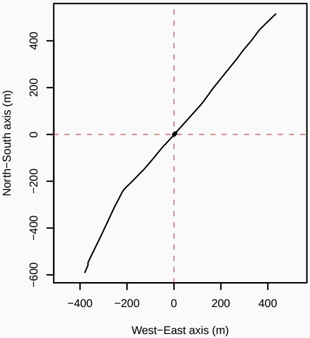
- Likelihood function $\ell(\mathbf{Y}_k | \mathbf{x}_k^{(i)}, \sigma_y^2, h, g)$
- Compute weights
- Weighted resample with replacement



Vehicle State Model: Particle Filter

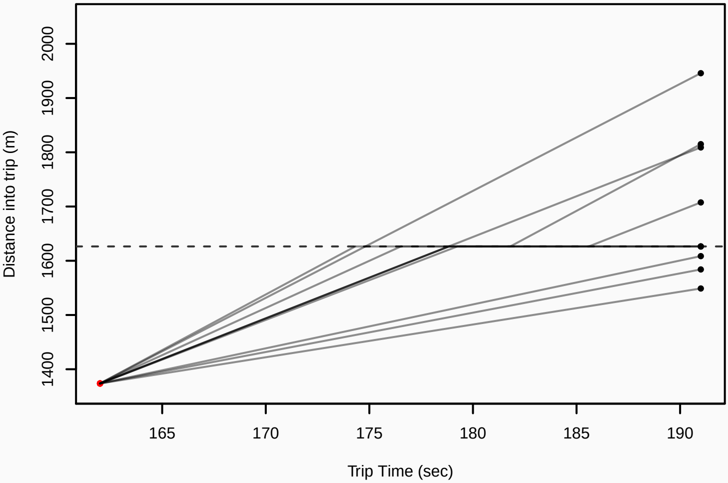
Step 2: update

- Likelihood function $\ell(\mathbf{Y}_k | \mathbf{x}_k^{(i)}, \sigma_y^2, h, g)$
- Compute weights
- Weighted resample with replacement
  $\Rightarrow$ keep particles plausible given data



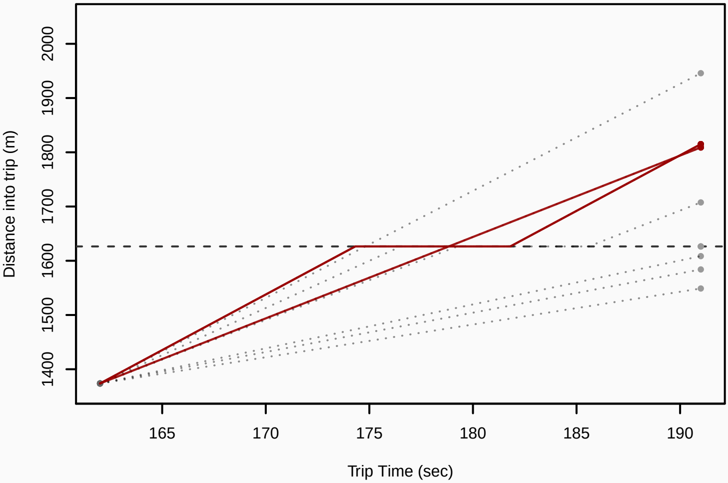
Vehicle State Model: Particle Filter

Step 2: update



Vehicle State Model: Particle Filter

Step 2: update



Road State Model

Road State Model

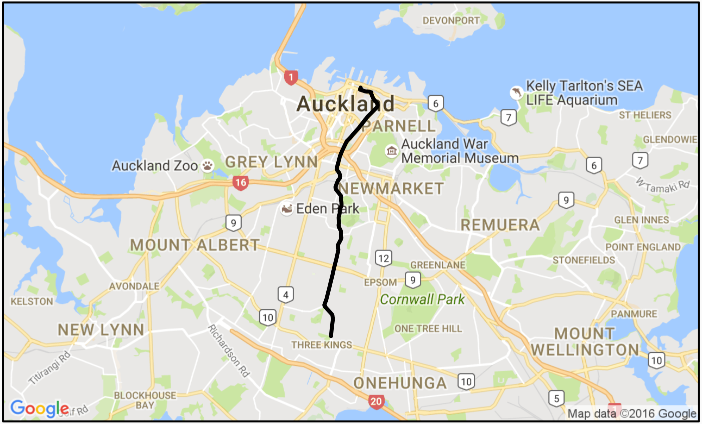
1. Particle filter $\Rightarrow$ speed estimates for a given bus
2. Identify segments of road common to multiple routes
3. Estimate speed along road segments using all busses

Road State Model

1. ~~Particle filter $\Rightarrow$ speed estimates for a given bus~~
2. Identify segments of road common to multiple routes
3. Estimate speed along road segments using all busses

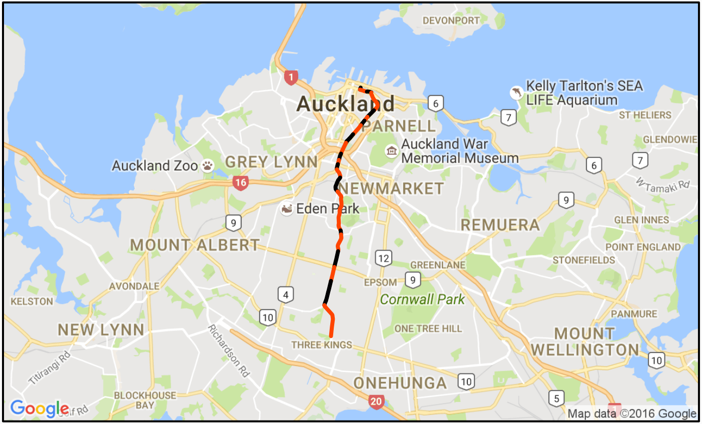
Road State Model

2. Identify segments of road common to multiple routes



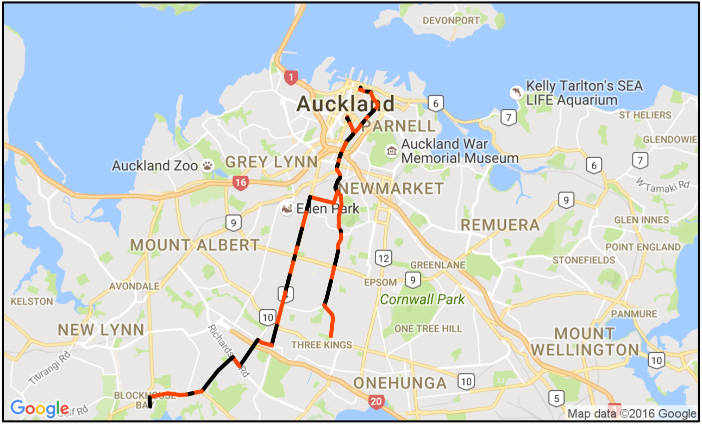
Road State Model

2. Identify segments of road common to multiple routes
   $\Rightarrow$ between (major) intersections



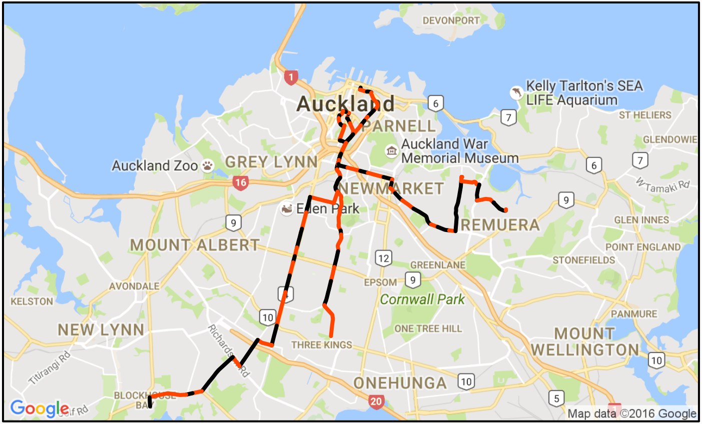
Road State Model

2. Identify segments of road common to multiple routes
   $\Rightarrow$ between (major) intersections



Road State Model

2. Identify segments of road common to multiple routes
   $\Rightarrow$ between (major) intersections



Road State Model

3. Estimate speed along road segments using <u>all busses</u>

Road State Model

3. Estimate speed along road segments using <u>all busses</u>

$\Rightarrow$ **Kalman filter**

Road state: mean speed for all M road segments at time t_ℓ

$$\boldsymbol{\nu}_\ell = [\nu_{1\ell}\ \nu_{2\ell}\ \cdots\ \nu_{M\ell}]^T$$

with associated covariance matrix

$$\Xi_\ell = \begin{bmatrix} \xi_{1\ell} & 0 & \cdots & 0 \\ 0 & \xi_{2\ell} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \xi_{M\ell} \end{bmatrix}$$

Road State Model

3. Estimate speed along road segments using <u>all busses</u>

⇒ **Kalman filter**

- no complex model necessary (Normal distribution adequate)

Road State Model

3. Estimate speed along road segments using all busses

⇒ **Kalman filter**

- no complex model necessary (Normal distribution adequate)
- updated using particle filter estimates

$$\mathbf{v}_\ell = \boldsymbol{\nu}_\ell + \mathbf{r}_\ell$$

- $\mathbf{v}_\ell$: mean speed of particles
- $\mathbf{r}_\ell \sim \mathcal{N}(0, \hat{\mathbf{R}}_\ell)$, $\hat{\mathbf{R}}_\ell$: variance of particle speeds

Predicting Arrival Time

Predicting Arrival Time

1. Schedule
2. Schedule deviation (AT?)
3. Vehicle state
4. Road state

Predicting Arrival Time

1. Schedule
2. Schedule deviation (AT?)
3. Vehicle state
4. Road state

Some notation:

- S_j^t = scheduled arrival time at stop j
- $\hat{A}_j$ = (predicted) arrival time at stop j
- $\tilde{T}_{s_k}^a, \tilde{T}_{s_k}^d$ = observed arrival/departure delay at last stop (from Auckland Transport's API)

Prediction Arrival Time

1. **Schedule**
   - $\hat{A}_j = S_j^t$
   - Baseline for other predictors

Prediction Arrival Time

2. Schedule deviation

- $\hat{A}_j = \begin{cases} S_j^t + \tilde{T}_{s_k}^d & \text{if departed stop } s_k \\ S_j^t + \tilde{T}_{s_k}^a & \text{if not departed stop } s_k \end{cases}$
- **OR** use particle estimates of arrival/departure delay, $\tilde{A}_{s_k}^{(i)}$ and $\tilde{D}_{s_k}^{(i)}$

Prediction Arrival Time

3. Vehicle state

- $S_j^d =$ distance along route of stop j

Prediction Arrival Time

3. Vehicle state

- S_j^d = distance along route of stop j
- $\hat{A}_j = t_k + \frac{S_j^d - d_k}{v_k}$

Prediction Arrival Time

3. Vehicle state

- S_j^d = distance along route of stop j
- $\hat{A}_j{}^{(i)} = t_k + \frac{S_j^d - d_k{}^{(i)}}{v_k{}^{(i)}}$
- Prediction for each particle

Prediction Arrival Time

3. Vehicle state

- $S_j^d =$ distance along route of stop j
- $\hat{A}_j{}^{(i)} = t_k + \frac{S_j^d - d_k{}^{(i)}}{v_k{}^{(i)}} + \sum_{\ell=s_k+1}^{j-1} p_\ell^{(i)} (\gamma + \bar{t}_\ell^{(i)})$
- Prediction for each particle
- Allow for dwell time uncertainty

Prediction Arrival Time

4. Road state

- r_k = route segment index
  R_b = distance along route of start of segment b

Prediction Arrival Time

4. Road state

- $r_k =$ route segment index
  $R_b =$ distance along route of start of segment b
- $\hat{A}_j = t_k +$

Prediction Arrival Time

4. Road state

- r_k = route segment index
  R_b = distance along route of start of segment b
- $\hat{A}_j = t_k + \frac{R_{s_k+1} - d_k}{\nu_{s_k}} +$
- travel time until end of current segment

Prediction Arrival Time

4. Road state

- r_k = route segment index
  R_b = distance along route of start of segment b
- $\hat{A}_j = t_k + \frac{R_{s_k+1} - d_k}{\nu_{s_k}} + \sum_{b \in B^\star} \frac{R_{b+1} - R_b}{\nu_b} +$
- travel time through intermediate segments ($B^\star$)

Prediction Arrival Time

4. Road state

- r_k = route segment index
  R_b = distance along route of start of segment b
- $\hat{A}_j = t_k + \frac{R_{s_k+1} - d_k}{\nu_{s_k}} + \sum_{b \in B^\star} \frac{R_{b+1} - R_b}{\nu_b} + \frac{S_j^d - R_{b'}}{\nu_{b'}}$
- travel time along segment b' to stop j

Prediction Arrival Time

4. Road state

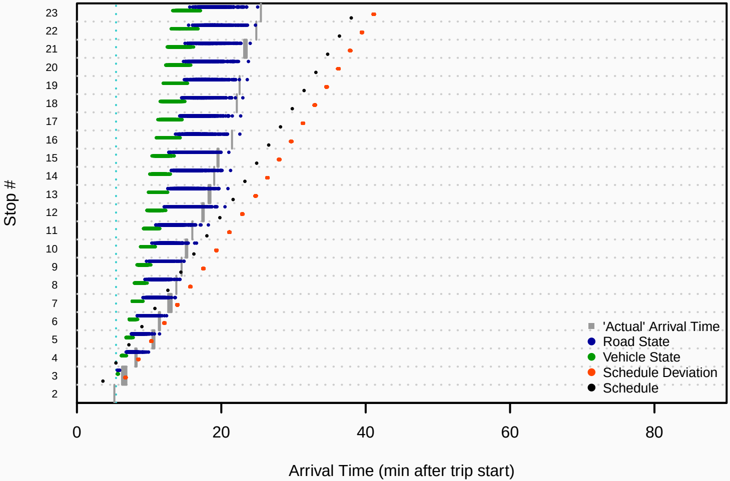
- r_k = route segment index
  R_b = distance along route of start of segment b
- $\hat{A}_j{}^{(i)} = t_k + \frac{R_{s_k{}^{(i)}+1} - d_k{}^{(i)}}{v_{s_k}^{(i)}} + \sum_{b \in B^\star} \frac{R_{b+1} - R_b}{v_b^{(i)}} + \frac{S_j^d - R_{b'}}{v_{b'}^{(i)}}$
- For each particle, sample $v_b^{(i)} \sim \mathcal{N}(\nu_b, \xi_b)$

Prediction Arrival Time

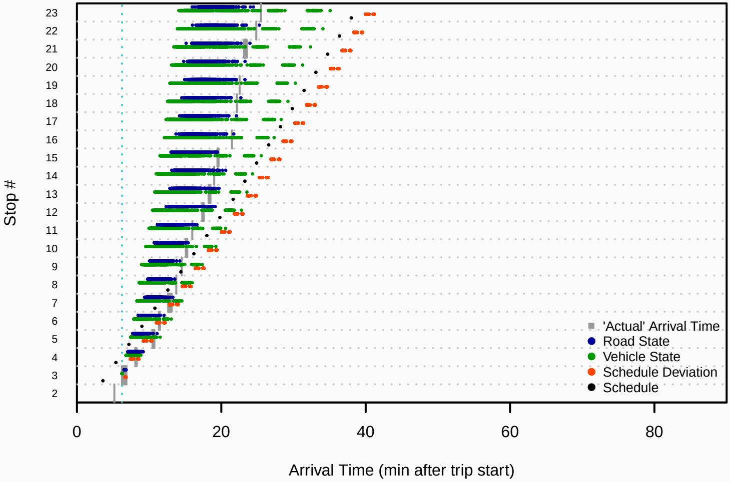
4. Road state

- r_k = route segment index
  R_b = distance along route of start of segment b
- $\hat{A}_j = t_k + \frac{R_{{s_k}^{(i)}+1} - {d_k}^{(i)}}{v_{s_k}^{(i)}} + \sum_{b \in B^\star} \frac{R_{b+1} - R_b}{v_b^{(i)}} + \frac{S_j^d - R_{b'}}{v_{b'}^{(i)}}$
  $+ \sum_{\ell = s_k + 1}^{j-1} p_\ell^{(i)} (\gamma + \bar{t}_\ell^{(i)})$
- For each particle, sample $v_b^{(i)} \sim \mathcal{N}(\nu_b, \xi_b)$
- Allow for dwell time uncertainty

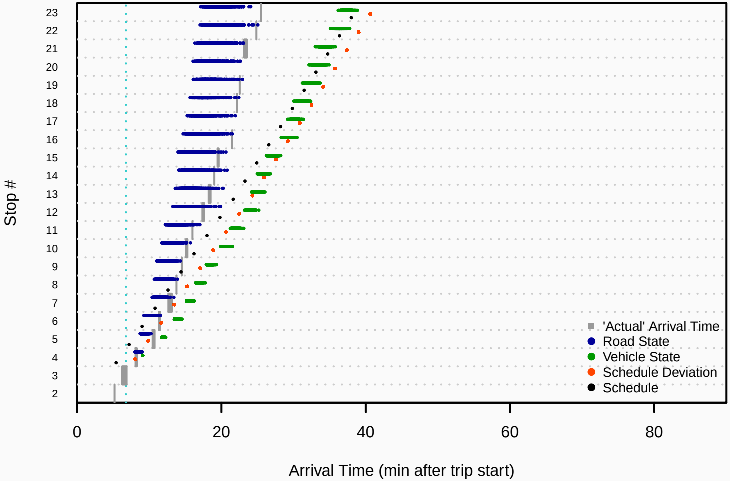
Predicting Arrival Time



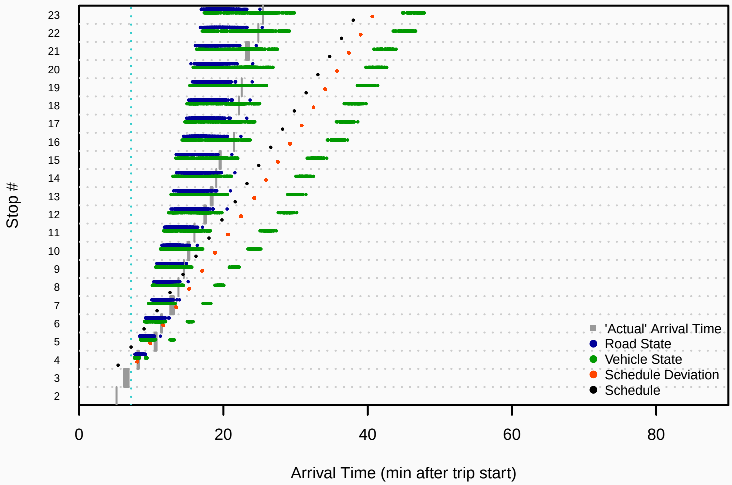
Predicting Arrival Time



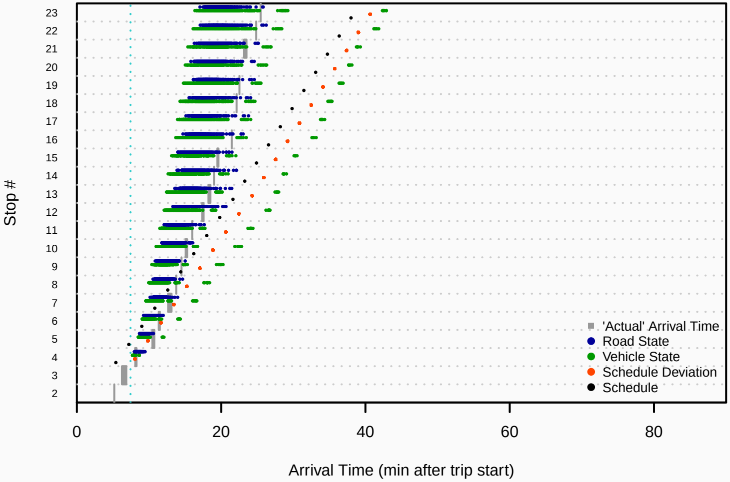
Predicting Arrival Time



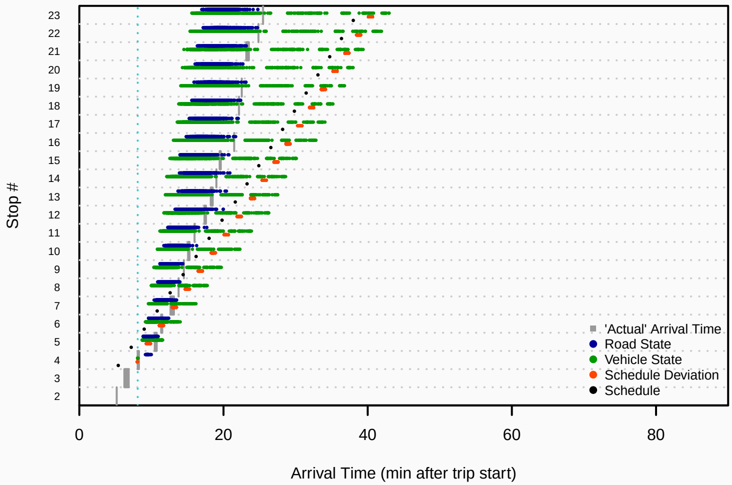
Predicting Arrival Time



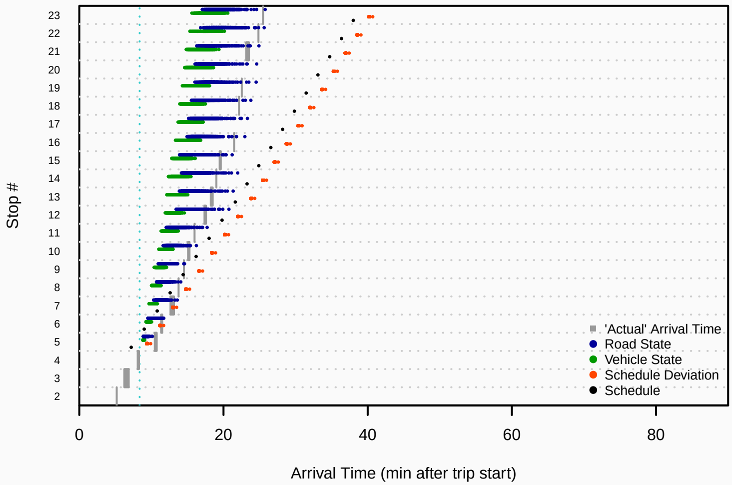
Predicting Arrival Time



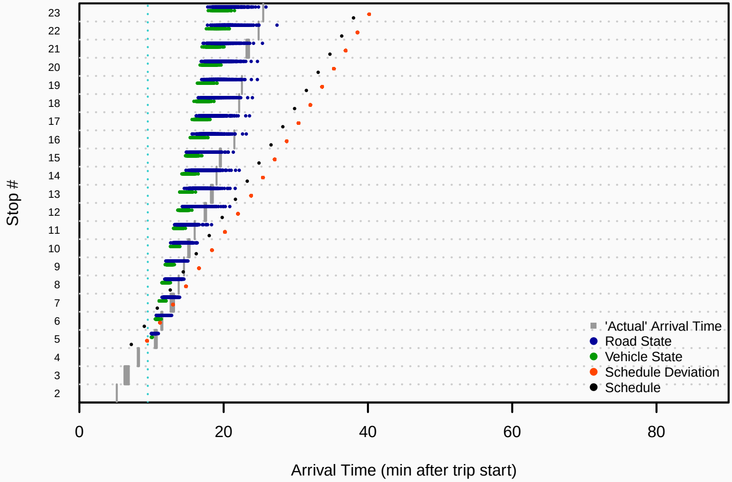
Predicting Arrival Time



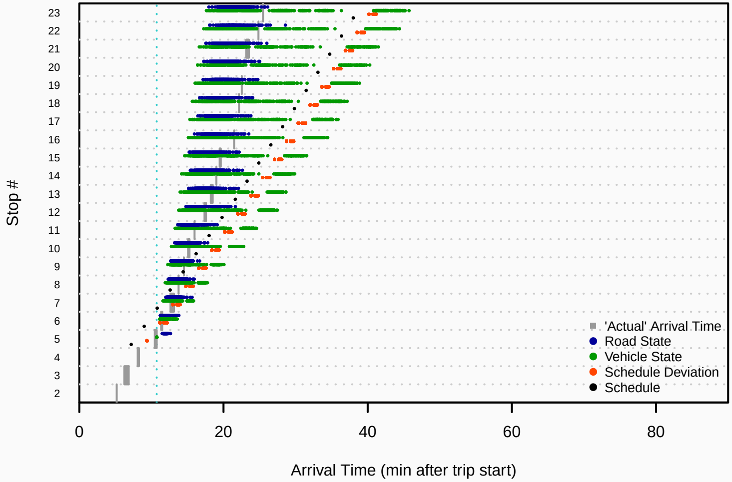
Predicting Arrival Time



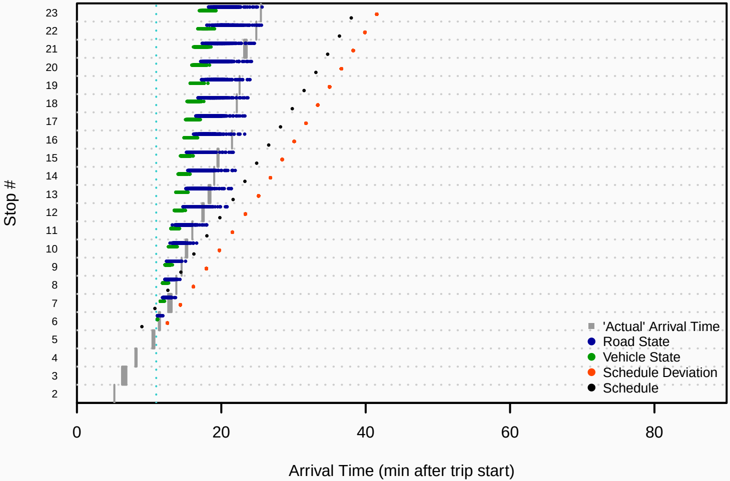
Predicting Arrival Time



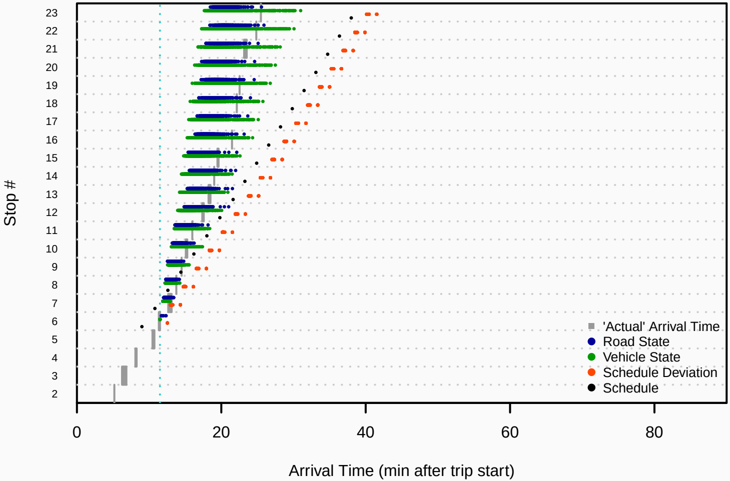
Predicting Arrival Time



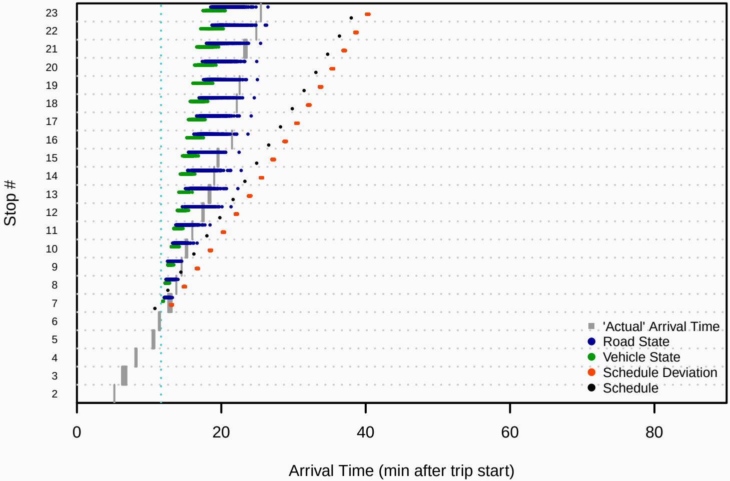
Predicting Arrival Time



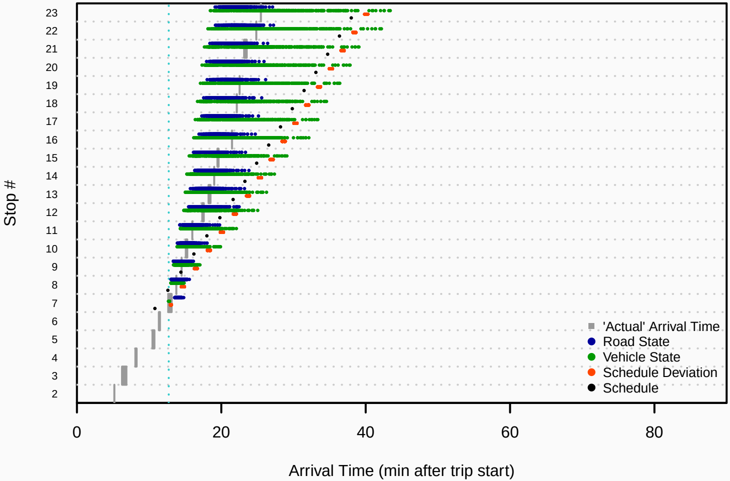
Predicting Arrival Time



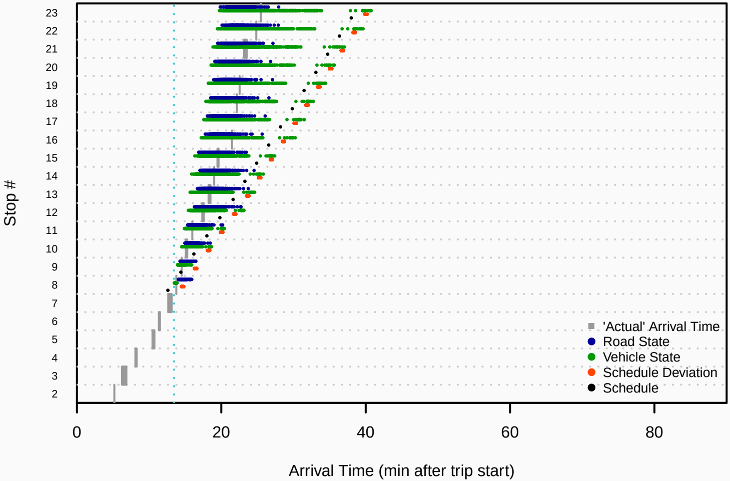
Predicting Arrival Time



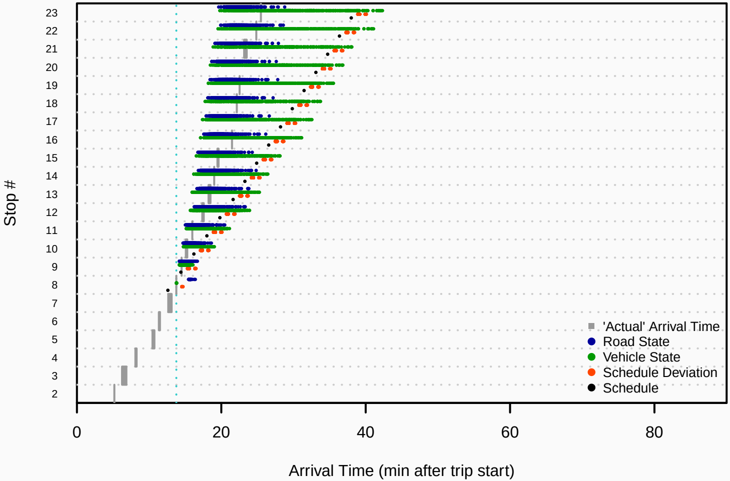
Predicting Arrival Time



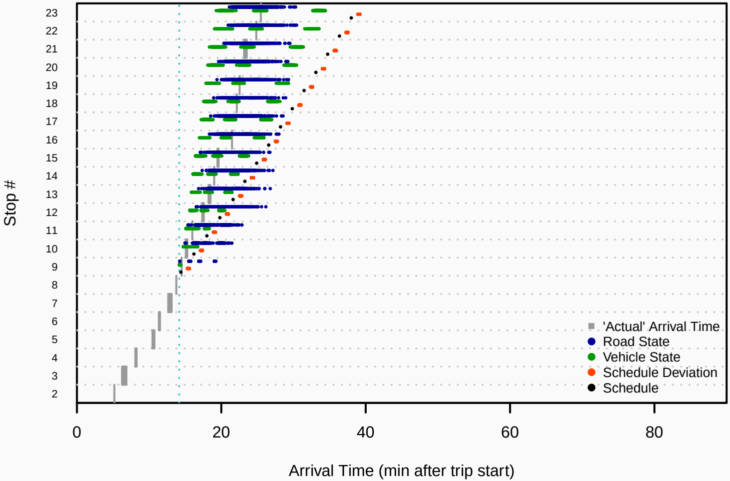
Predicting Arrival Time



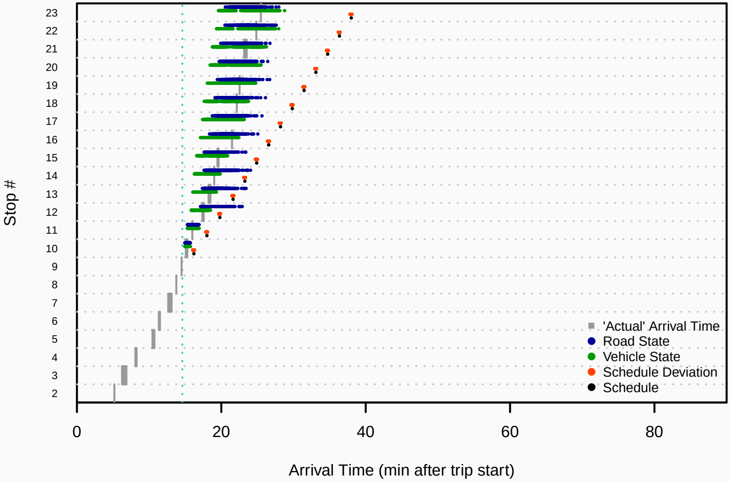
Predicting Arrival Time



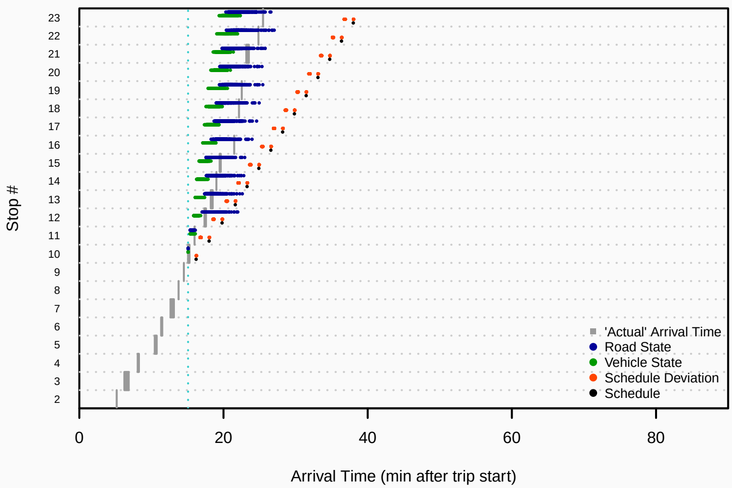
Predicting Arrival Time



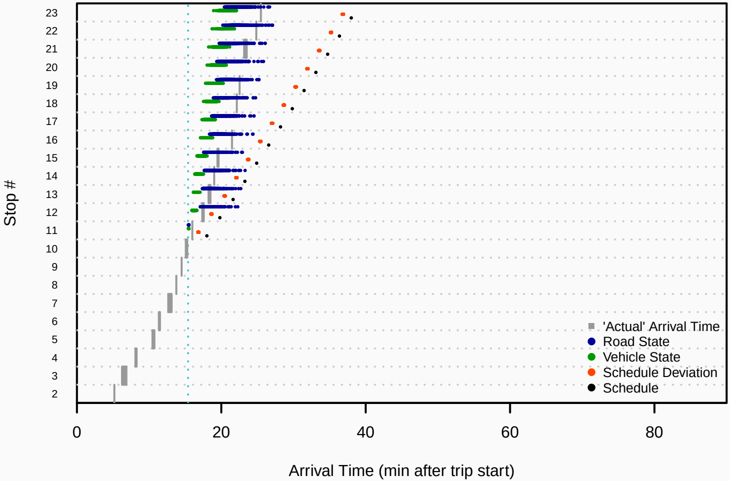
Predicting Arrival Time



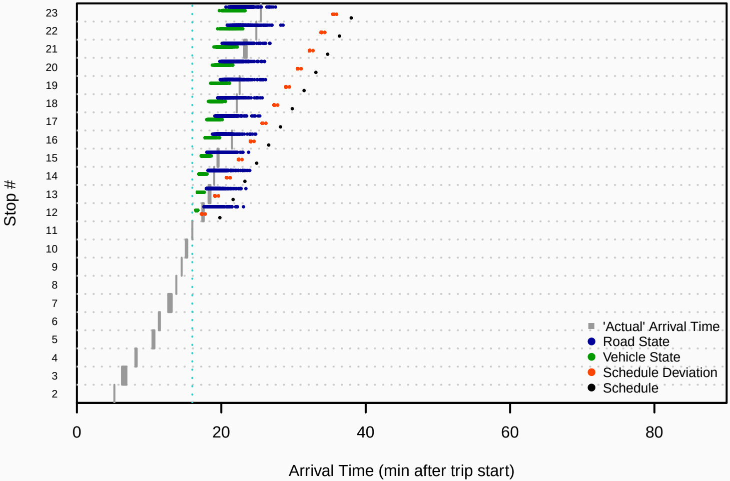
Predicting Arrival Time



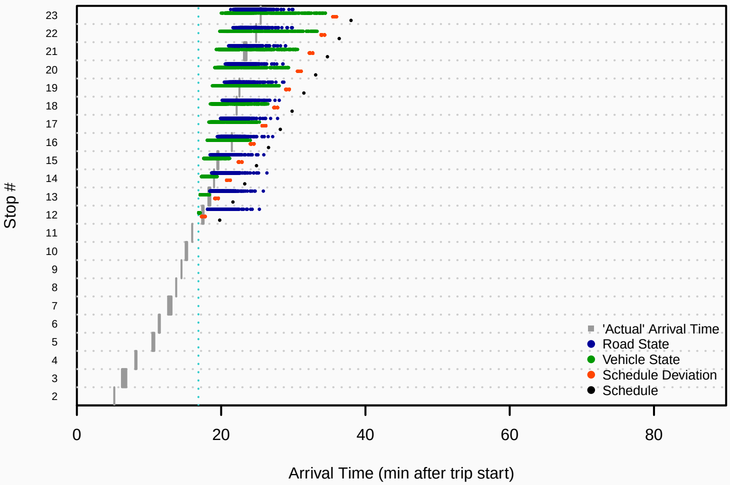
Predicting Arrival Time



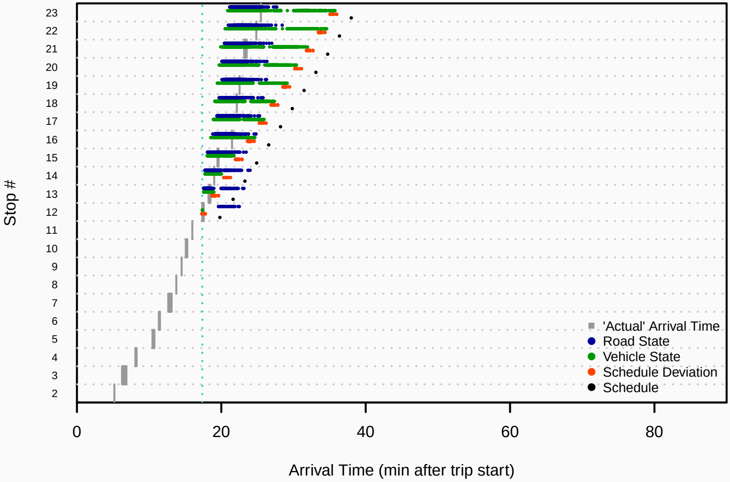
Predicting Arrival Time



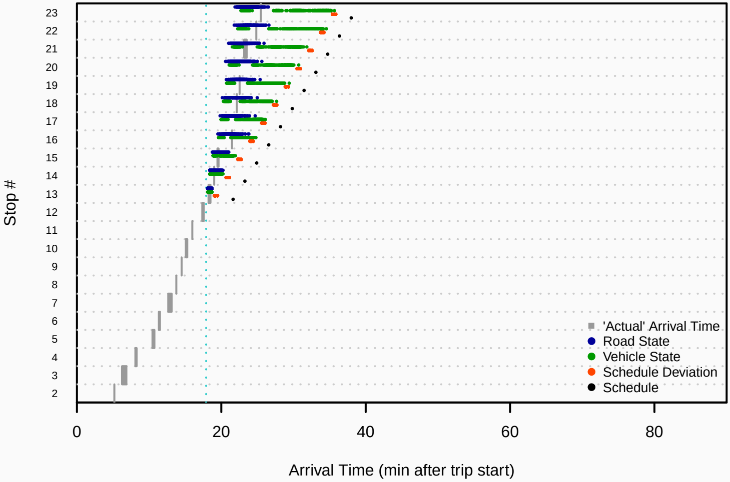
Predicting Arrival Time



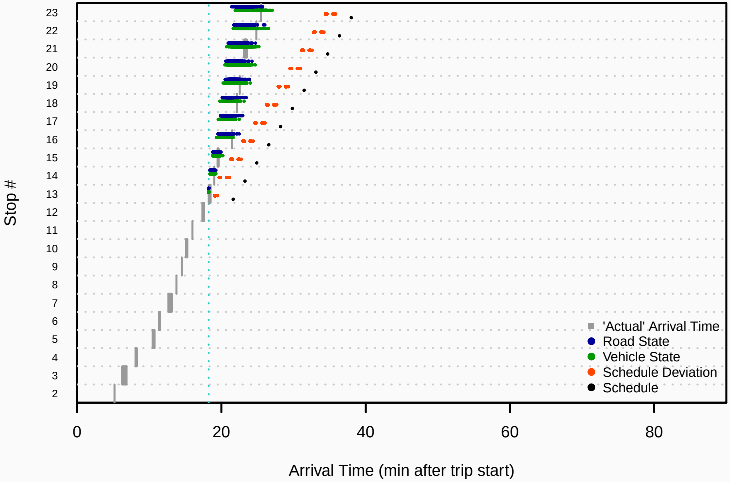
Predicting Arrival Time



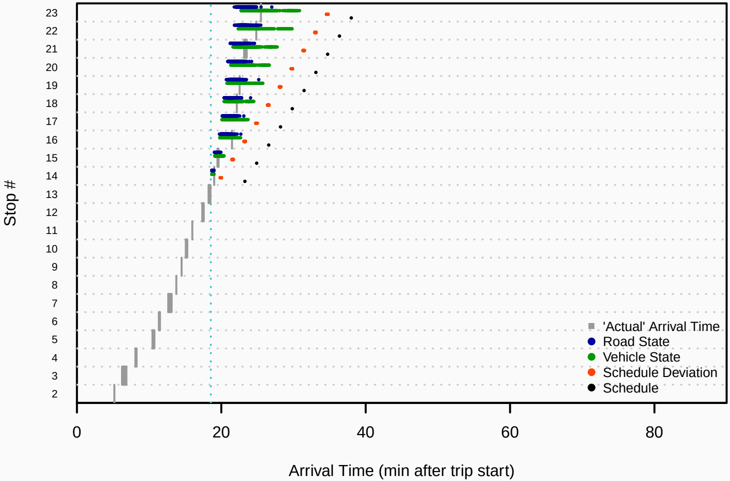
Predicting Arrival Time



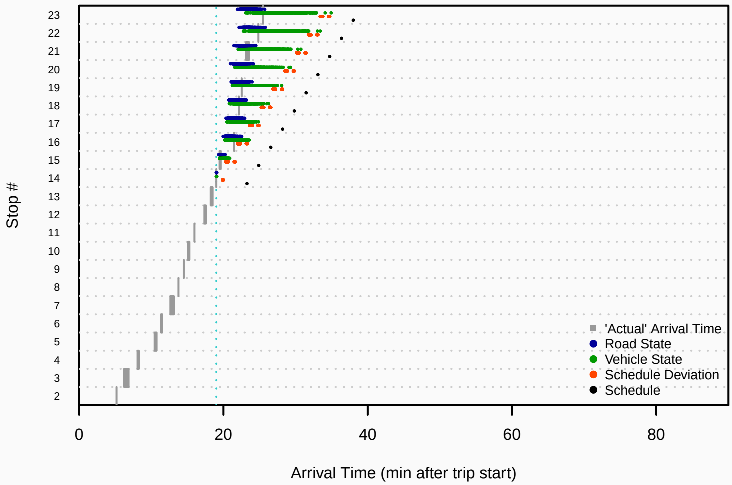
Predicting Arrival Time



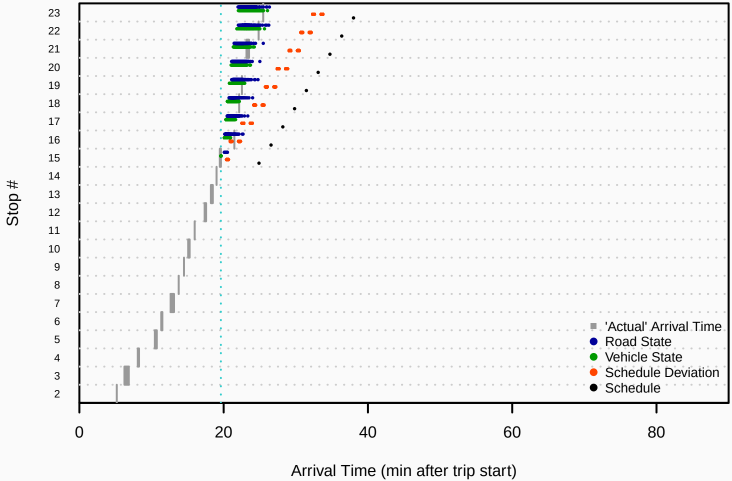
Predicting Arrival Time



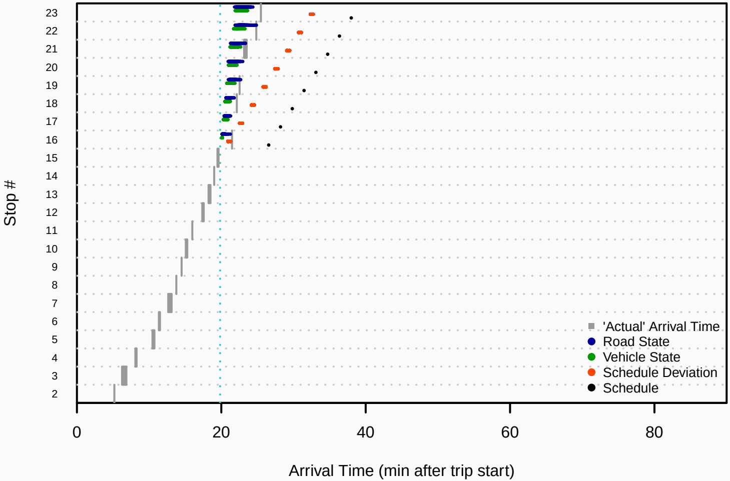
Predicting Arrival Time



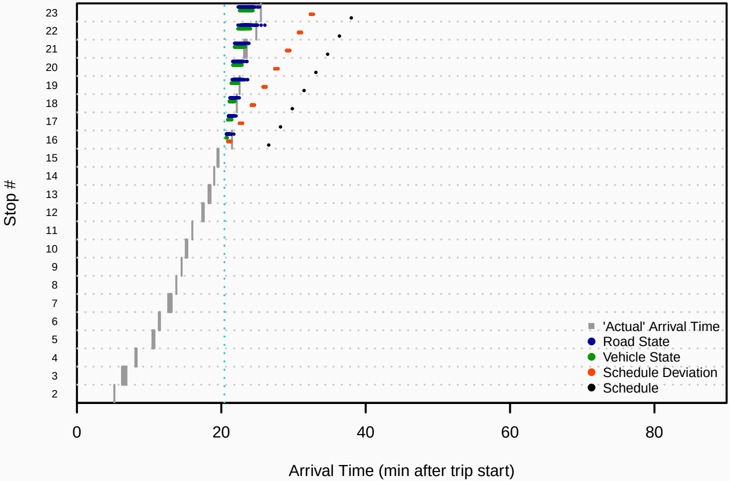
Predicting Arrival Time



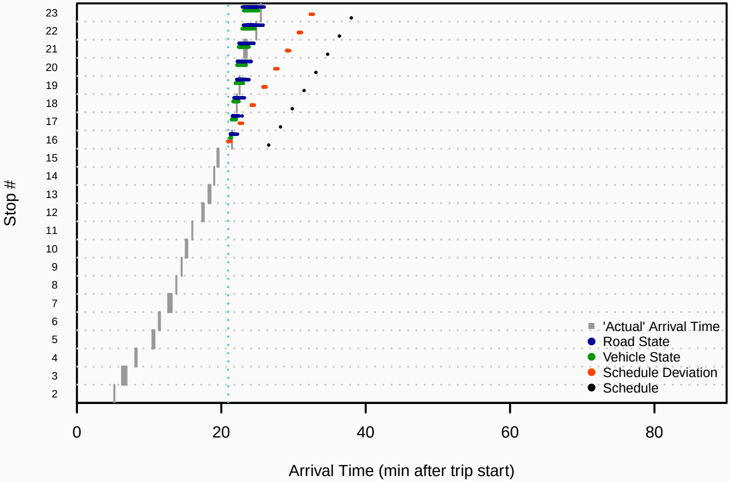
Predicting Arrival Time



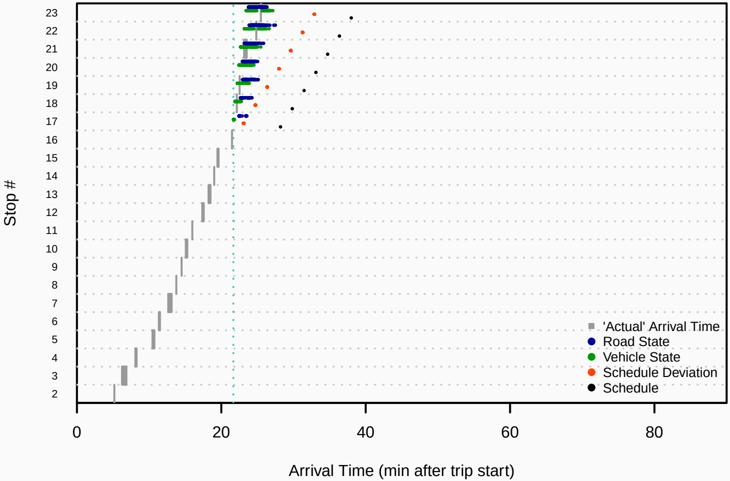
Predicting Arrival Time



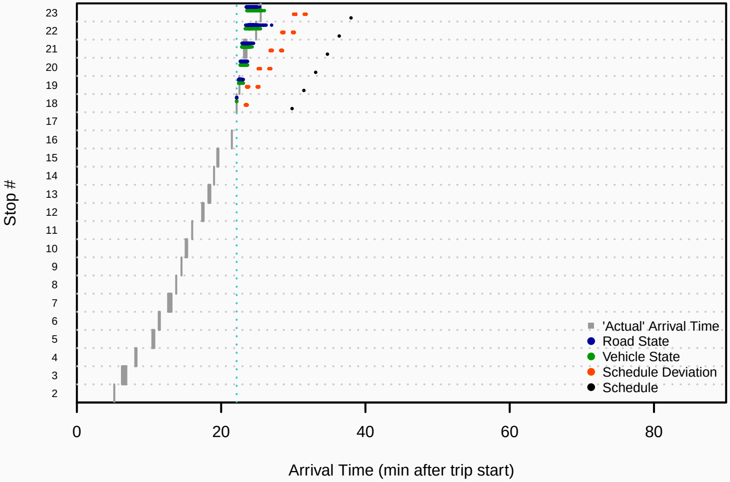
Predicting Arrival Time



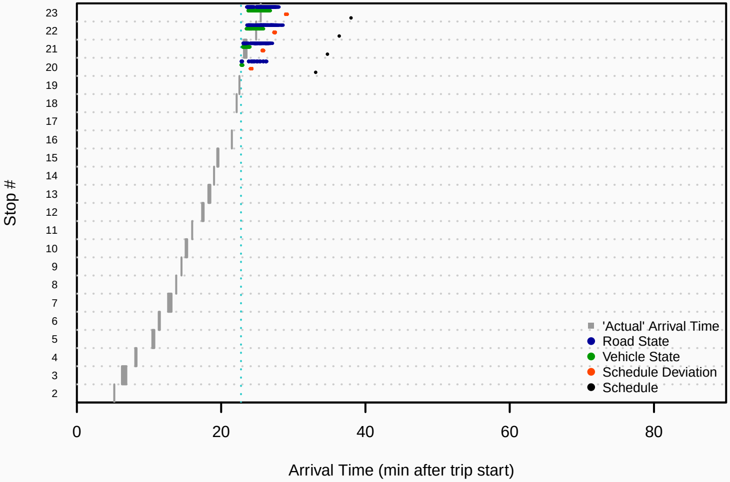
Predicting Arrival Time



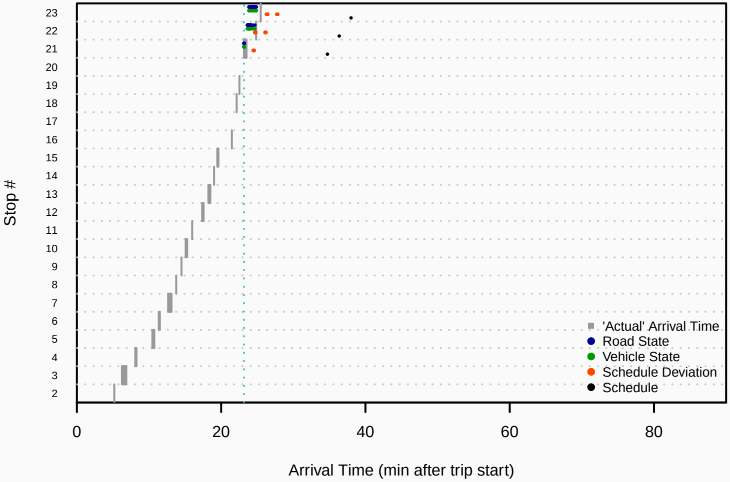
Predicting Arrival Time



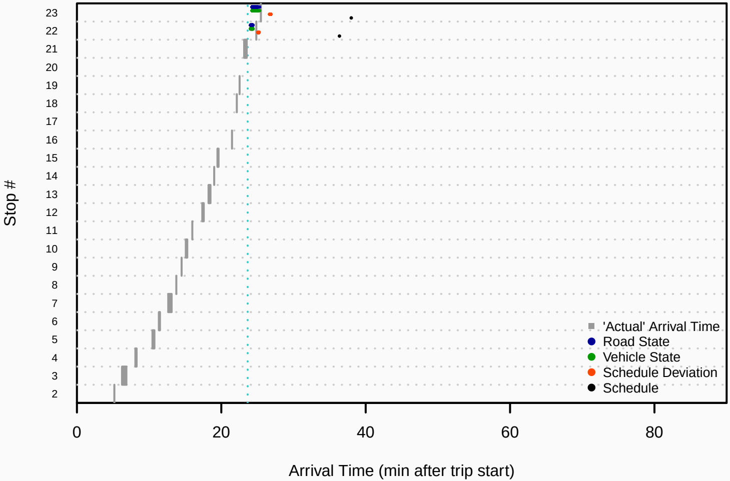
Predicting Arrival Time



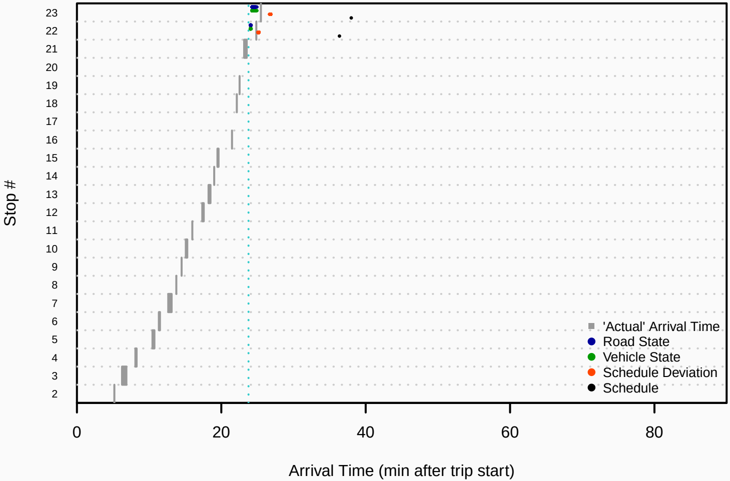
Predicting Arrival Time



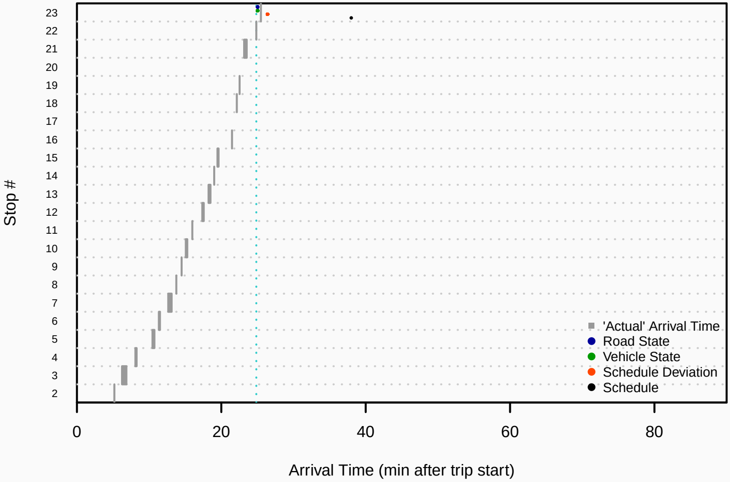
Predicting Arrival Time



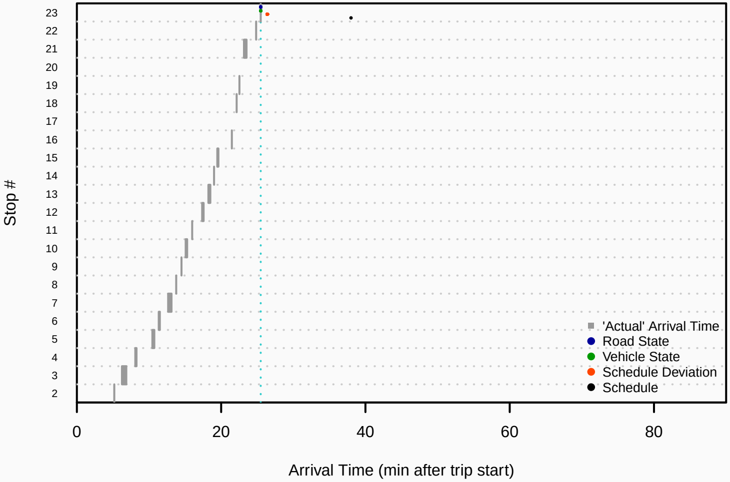
Predicting Arrival Time



Predicting Arrival Time



Predicting Arrival Time



Predicting Arrival Time

Conclusions:

- **Schedule:** ...

Predicting Arrival Time

Conclusions:

- **Schedule:** ...
- **Schedule deviation:** OK for very short-range prediction; relies on time table accuracy

Predicting Arrival Time

Conclusions:

- **Schedule:** ...
- **Schedule deviation:** OK for very short-range prediction; relies on time table accuracy
- **Vehicle state:** high variability; OK for short-range prediction

Predicting Arrival Time

Conclusions:

- **Schedule:** ...
- **Schedule deviation:** OK for very short-range prediction; relies on time table accuracy
- **Vehicle state:** high variability; OK for short-range prediction
- **Road state:** little variability; performs well at long-range prediction

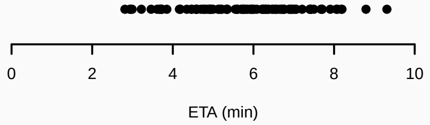
Predicting Arrival Time

Conclusions:

- **Schedule:** ...
- **Schedule deviation:** OK for very short-range prediction; relies on time table accuracy
- **Vehicle state:** high variability; OK for short-range prediction
- **Road state:** little variability; performs well at long-range prediction

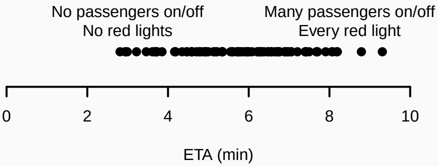
Predicting Arrival Time: Intervals

How do we communicate estimate + uncertainty to commuters?



Predicting Arrival Time: Intervals

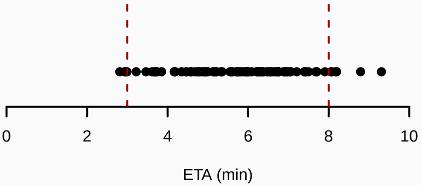
How do we communicate estimate + uncertainty to commuters?



Predicting Arrival Time: Intervals

How do we communicate estimate + uncertainty to commuters?

⇒ **Prediction intervals**

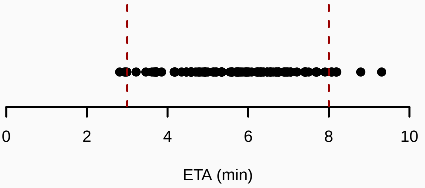


Predicting Arrival Time: Intervals

How do we communicate estimate + uncertainty to commuters?

⇒ **Prediction intervals**

- easy to compute from particle sample
- (arguably) intuitive: ETA 6 min (mean) versus ETA 3–8 min
- Biased to reduce chance of missing bus



What's Next?

- Add more routes
  - $\Rightarrow$ automate intersection detection
- Historical data to estimate parameters
  - $\Rightarrow$ Dwell times, stopping probabilties
  - $\Rightarrow$ Segment speed covariance matrix (including off-diagonals)
  - $\Rightarrow$ Model wait time at intersections
- Scale up: ALL routes/busses
  - $\Rightarrow$ computational speed
  - $\Rightarrow$ run in real-time
- Selection of "best" quantiles for prediction intervals
- ...

Thank you!

Questions?