How can we improve bus ETAs?
Using real-time position data to estimate road state

Tom Elliott and Thomas Lumley

Department of Statistics, University of Auckland, New Zealand

tom.elliott@auckland.ac.nz

1. Introduction

- real time information (RTI) has become an important component of public transit systems, providing commuters with up to date information on the location and estimated arrival time (ETA) of buses
- transit vehicle tracking has been used for several decades to obtain ETAs, with much research into methods for improving the vehicle models used [1–4]
- the main source of uncertainty is congestion, particularly in networks with poor public tranport infrastructure (e.g., dedicated bus lanes)
- research shows that ETA accuracy can be improved by using travel time information from previous buses along a road [5], however this only applied to a specific, manually selected road
- we propose a generalised approach to modeling transit vehicles and network congestion
  - a vehicle model estimates vehicle speed/travel time along a road
  - the state (vehicle speed) of the road is updated to reflect real-time congestion
  - ETAs are updated using congestion information along all intermediate roads

2. GTFS network construction

- **GTFS** is an API specification for transit data [6], and includes route shape information
- available in over 500 locations worldwide
- **transit network** consisting of intersections and connecting road segments constructed from the raw GTFS data

1. raw GTFS data provides **one shape per route**
2. **identify points of intersection** between one or more routes using algorithm adapted from [7]
3. **split shapes at intersections** to obtain shapes for each individual road segment
4. express each route as a sequence of road segments

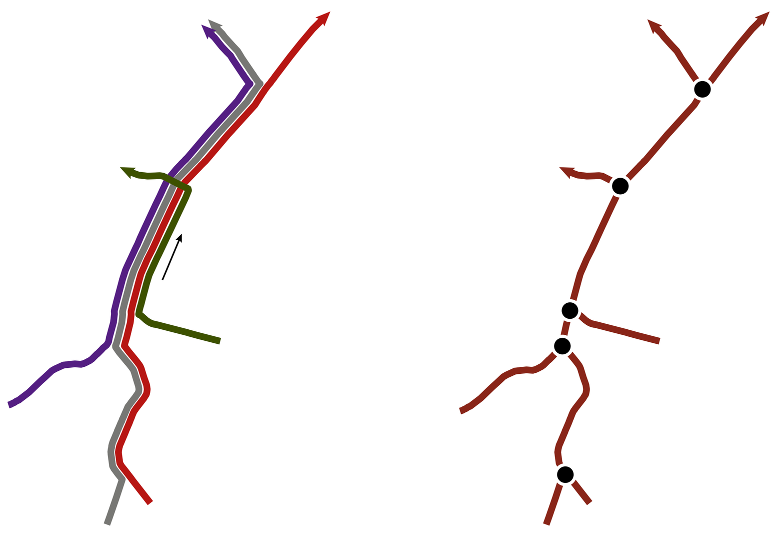


Figure 1: *An example transit netork produced from five routes. Left: the raw GTFS shapes; Right: the generated transit network with intersections shown as dots.*

- Implementation in progress: the `gtfsnetwork` R package, github.com/tmelliott/gtfsnetwork

3. Vehicle state model

- sequential Bayesian methods well suited to **real-time vehicle tracking**
- in Auckland, GTFS realtime positions obtained approximately every 30 seconds, but high variability
- **particle filter**: general, flexible estimation method
  - vehicle state approximated by a sample of particles $\tilde{\boldsymbol{X}}_k = (\boldsymbol{X}_k^{(i)})_{i=1}^N$
  - particles transitioned independently, so multimodal distributions are no issue (e.g., at bus stops and intersections)
  - likelihood is intuitive: the distance between the estimated and observed location
- measurement function $h : \mathbb{R} \mapsto \mathbb{R}^2$ calculates particle's map position from distance traveled along shape
- our particle filter estimation procedure is as follows:

1. predict trajectory of each particle using transition function f and system noise parameter Q_k as shown in figure 3(a)
$$\boldsymbol{X}_k^{(i)} = f(\boldsymbol{X}_k^{(i)}, w_k), \quad w_k \sim N(0, Q_{k-1})$$
2. assume $\boldsymbol{Y}_k$ is a noisy measurement of true position with GPS error σ_y^2 (fig 2), and define $g : \mathbb{R}^2 \mapsto \mathbb{R}^2$ such that $dist(g(\boldsymbol{Y}_1), g(\boldsymbol{Y}_2))$ is the ground distance between the points, then the measurement model is
$$g(\boldsymbol{Y}_k) \sim N\left(g(h(\boldsymbol{X}_k)), \begin{bmatrix}\sigma_y^2 & 0 \\ 0 & \sigma_y^2\end{bmatrix}\right)$$
and $(\delta_k^{(i)})^2 = dist(g(h(\boldsymbol{X}_k^{(i)})), g(Y_k))^2$ is the sum of two independent normal r.v.'s with mean 0 and variance σ_y^2
$$\left((\delta_k^{(i)})^2/\sigma_y^2\right) \sim \chi^2(2) \sim \text{Exp}(0.5)$$
3. evaluated the likelihood for each particle
$$p(\boldsymbol{Y}_k | \boldsymbol{X}_k^{(i)}) = 0.5 e^{-(\delta_k^{(i)})^2 / 2\sigma_y^2}$$
4. update state by resampling particles with replacement, using likelihood weights (fig 3(b))
$$w^{(i)} = \frac{p(\boldsymbol{Y}_k | \boldsymbol{X}_k^{(i)})}{\sum_{j=1}^N p(\boldsymbol{Y}_k | \boldsymbol{X}_k^{(j)})}$$
5. use resulting trajectories to estimate vehicle speed along road segments to update network in section 4
   - v_k^j is the mean speed of the particles, and
   - e_k^j is the variance of particle speeds for a vehicle

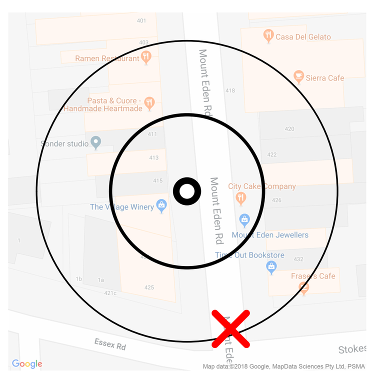
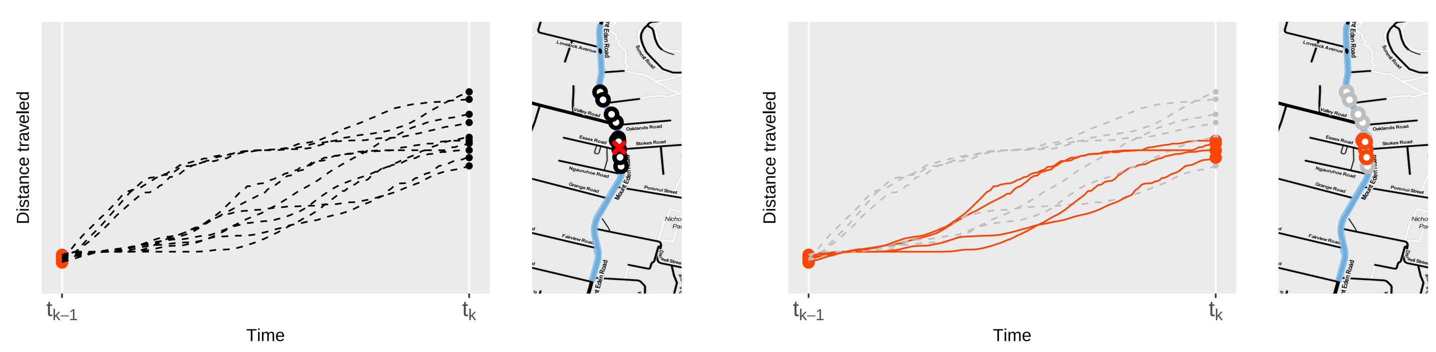


Figure 2: *Y_k (red cross) is a bivariate normal r.v. with mean and variance represented by the black dot and concentric rings, respectively.*



(a) Forecasted particle trajectories (b) Updated particle sample are resampling

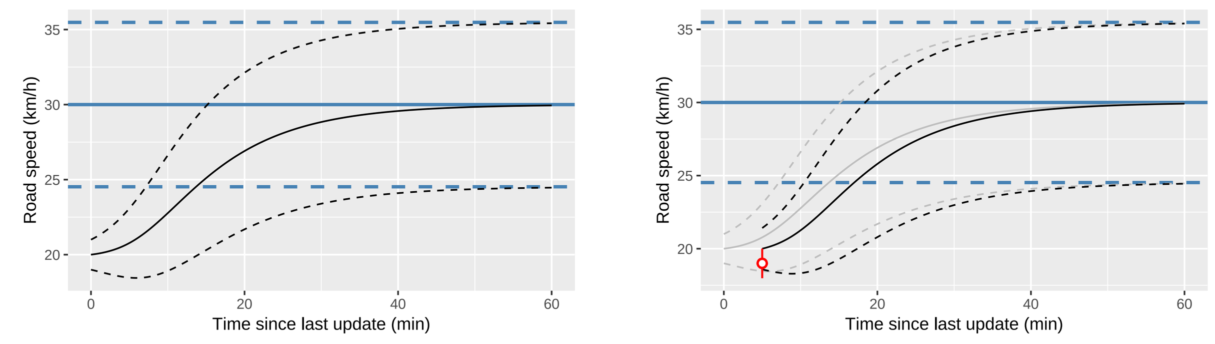
Figure 3: *The vehicle state consists of distance traveled (the y-axes in the graphs). The predicted state for each particle involves a trajectory (a), and the final position can be mapped to a location on a map, which makes comparing the predictions to the observation (in red) intuitive and straightforward.*

4. Network state model

- goal is to develop a model that can estimate **real-time** and **future** network state for use in arrival time prediction
- **network state** is defined as the average speed β_r^j of buses along road segment j at time t_r
- historical data is used to determine the prior mean and variance (blue lines in figure 4(b))
$$p(\beta_r^j) \sim N(\mu_j(t_r), \psi_j(t_r)^2)$$
- parameters $\hat{\beta}_r^j$ and P_r^j are the state estimate and variance, respectively, estimated using an adapted **extended Kalman filter (EKF)** algorithm
$$p(\beta_r^j | v_r^j) \sim N(\hat{\beta}_r^j, P_r^j)$$

1. predict future state, fig. 4(a)
   - **transition function** and **system noise** defined such that $\hat{\beta}_r^j$ and P_r^j **converge** to $\mu_j(t_r)$ and $\psi_j(t_r)$, respectively
   - EKF predict equations used to recursively estimate state in one second intervals
2. update state, fig. 4(b)
   - use values from step 5 in section 3
   - EKF update equations with observation v_r^j and measurement error e_r^j, respectively

- *Here, we are assuming a constant speed along the length of each road; we are currently working on a model that allows variable speeds.*
- this transition function is also used in arrival time prediction (section 5)
  - takes time for vehicles to travel from current postiion to road segments further down the route
  - ETAs need to account for predicted congestion behaviour (for example before/after peak hour), which is defined in the prior



(a) Forecasted network state (b) Updated and re-forecasted state

Figure 4: *Network road state predictions as defined by the transition function converge towards the prior mean and variance (in blue) over time. These state forecasts are used for predicting future travel times for arrival time prediction, and as the state prediction for the update step that occurs whenever a bus travels along the road segment (in red).*

5. Predicting arrival time

- accurate arrival time predictions need to consider **current vehicle location**, **current traffic state**, as well as **future traffic states**, traffic lights, and intermediate bus stop wait times
- for each particle, simulate journey along remainder of route
  - simulate particle speed from network model (normal r.v.) at time of arrival at each road segment in sequence
  - introduce wait times (exponential) at intersections and bus stops
  - calculate arrival times at each upcoming stop
- yeilds a distribution of arrival times for each stop (fig 5)
- ETAs are typically reported in discrete minutes. For example, the distribution in figure 5 might be summarised with
  - a **point estimate** of 5 minutes
  - a **prediction interval** of 4–8 minutes
- summary statistics need to be chosen such that, as the bus approaches, the estimates decrease, but also **minimising the probability that the bus arrives sooner than the ETA**

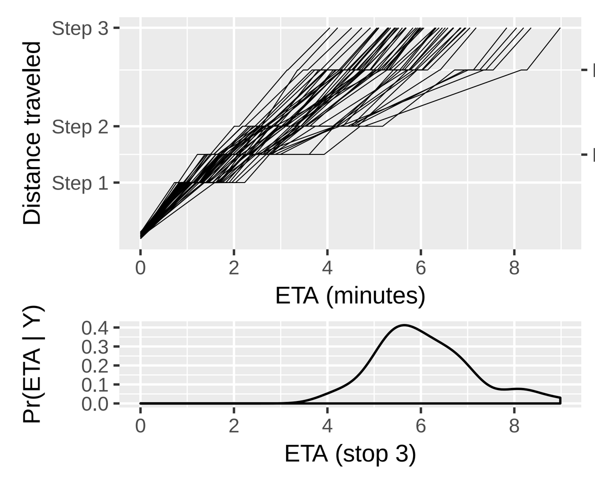


Figure 5: *Top: travel time predictions for a bus, showing locations of stops (left axis) and intersections (right axis). Bottom: posterior density of ETAs for stop 3.*

6. Conclusion and future work

- segmenting routes into route-independent segments allows vehicle observations to update the road network
- real-time network state used to predict arrival time
- current real-time C++ implementation takes up to 20 seconds on an 8-core Virtual Machine with 5000 particles per vehicle

Next steps:

- improve the network state model: **non-constant speeds** along a roads (i.e., $\mu_j(t, d)$ depends on time and distance along segment), include **covariates in state transition** (adjacent segments, yesterday's traffic, weather, etc.)
- develop a stop- and intersection-wait time model to more accurately simulate wait times
- investigate ideal summary statistics for ETAs (both point and interval prediction)

References

[1] Z. Wall and D.J. Dailey. An algorithm for predicting the arrival time of mass transit vehicles using automatic vehicle location data. In *Proceedings of the Transportation Research Board Annual Meeting*, 1999.

[2] D. Dailey, S. Maclean, F. Cathey, and Z. Wall. Transit vehicle arrival prediction: Algorithm and large-scale implementation. *Transportation Research Record: Journal of the Transportation Research Board*, 1771:46–51, jan 2001.

[3] F. W. Cathey and D. J. Dailey. A prescription for transit arrival/departure prediction using automatic vehicle location data. *Transportation Research Part C: Emerging Technologies*, 11(3-4):241–264, jun 2003.

[4] Etienne Hans, Nicolas Chiabaut, Ludovic Leclercq, and Robert L. Bertini. Real-time bus route state forecasting using particle filter and mesoscopic modeling. *Transportation Research Part C: Emerging Technologies*, 61:121–140, dec 2015.

[5] Bin Yu, William H. K. Lam, and Mei Lam Tam. Bus arrival time prediction at bus stop with multiple routes. *Transportation Research Part C: Emerging Technologies*, 19(6):1157–1170, dec 2011.

[6] Google Developers. What is GTFS? https://developers.google.com/transit/gtfs/, 2006.

[7] Yongchuan Zhang, Jiping Liu, Xinlin Qian, Agen Qiu, and Fuhao Zhang. An automatic road network construction method using massive gps trajectory data. *ISPRS International Journal of Geo-Information*, 6(12), 2017.

2018 ISBA World Meeting, Edinburgh, UK, June 24–29 2018