Improving bus arrival-time estimates

using real-time vehicle positions to estimate road state

Tom Elliott and Thomas Lumley

Department of Statistics, University of Auckland, New Zealand

`tom.elliott@auckland.ac.nz`

1. Introduction

- position tracking well studied, e.g., Kalman Filter [1–3], particle filter [4], etc.
- estimating and predicting road state (i.e., travel time along roads) less developed, particularly for bus prediction
- several papers use other vehicles *on the same route* [5], but no generic attempt to model travel times independently of route
- many public transport providers don't use any form of traffic model, instead relying on scheduled stop times (often inaccurate, don't respond to real-time events)

2. GTFS network construction

- **GTFS** is an API specification for transit APIs [6]
- available in over 500 locations worldwide
- therefore, a general approach to 'segmenting' the network is required

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

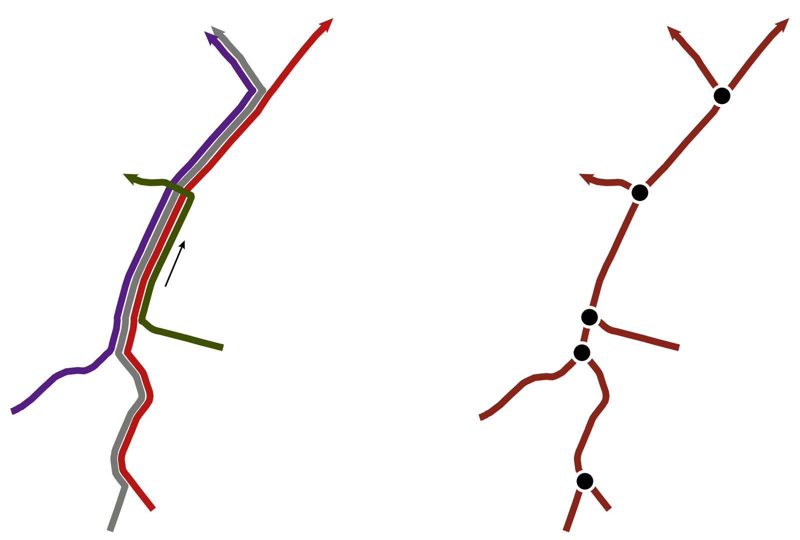


Figure 1: *An example transit netork produced from five routes. Left: the raw GTFS shapes; Right: the generated transit network with intersections shown as dots.*

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

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.

3. Vehicle state model

- sequential Bayesian methods well suited to **real-time vehicle tracking**
- **particle filter**: general, flexible estimation method that uses sample of particles $\tilde{\boldsymbol{X}}_k = (\boldsymbol{X}_k^{(i)})_{i=1}^N$, allowing it to handle **multimodality** (e.g., when passing bus stops) and **assymetry** (e.g., bus cannot go backwards)
- measurement function $h : \mathbb{R} \mapsto \mathbb{R}^2$ calculates map (GPS) position of each particle based on distance traveled along shape

1. predict trajectory of each particle using transition function f and system noise parameter Q_k

$$\boldsymbol{X}_k^{(i)} = f(\boldsymbol{X}_k^{(i)}, w_k), \quad w_k \sim N(0, Q_{k-1})$$

2. assume $\boldsymbol{Y}_k$ is a noisy measurement of true position with GPS error σ_y^2, 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

$$\left((\delta_k^{(i)})^2/\sigma_y^2\right) \sim \chi^2(2) \sim \mathrm{Exp}(0.5)$$

3. evaluated the likelihood for each particle

$$p(\boldsymbol{Y}_k|\boldsymbol{X}_k^{(i)}) = 0.5e^{-(\delta_k^{(i)})^2/2\sigma_y^2}$$

4. update state by resampling particles with replacement, using likelihood weights

$$w^{(i)} = p(\boldsymbol{Y}_k|\boldsymbol{X}_k^{(i)}) / \sum_{j=1}^{N} p(\boldsymbol{Y}_k|\boldsymbol{X}_k^{(j)})$$

5. use resulting trajectories to estimate vehicle speed along road segments to update network in section 4

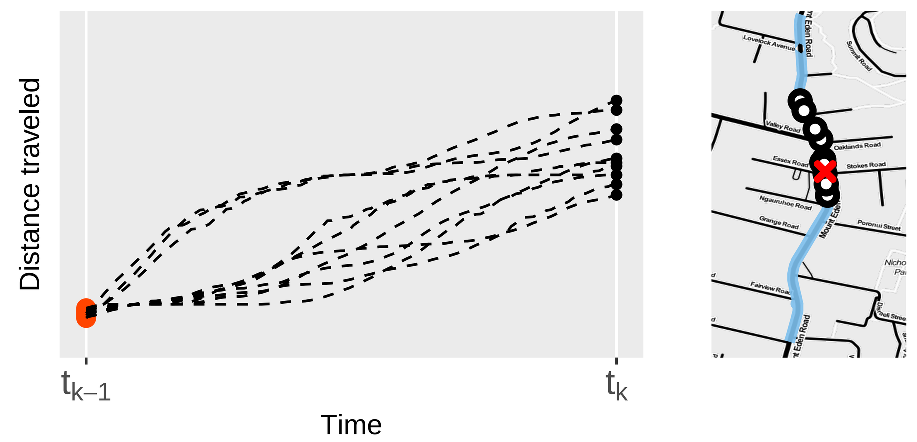


Figure 2: *Left: simulated particle trajectories. Right: particle positions $h(\boldsymbol{X}_k^{(i)})$; observation $\boldsymbol{Y}_k$ in red.*

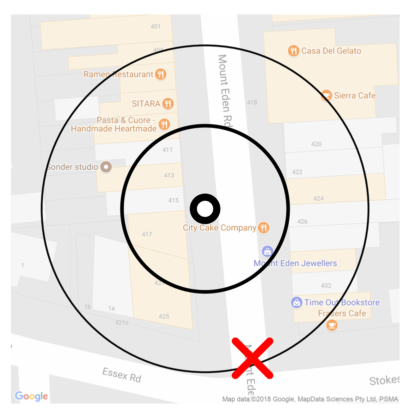


Figure 3: $\boldsymbol{Y}_k$ *(red cross) is a bivariate normal r.v. with mean and variance represented by the black dot and concentric rings, respectively.*

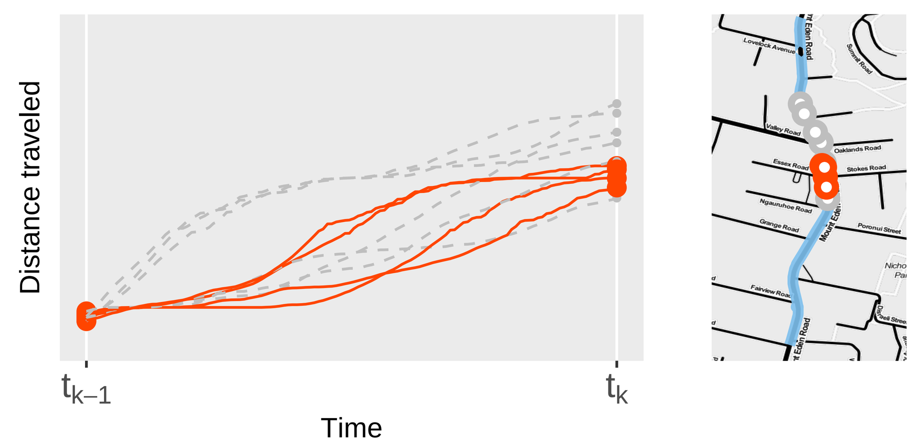
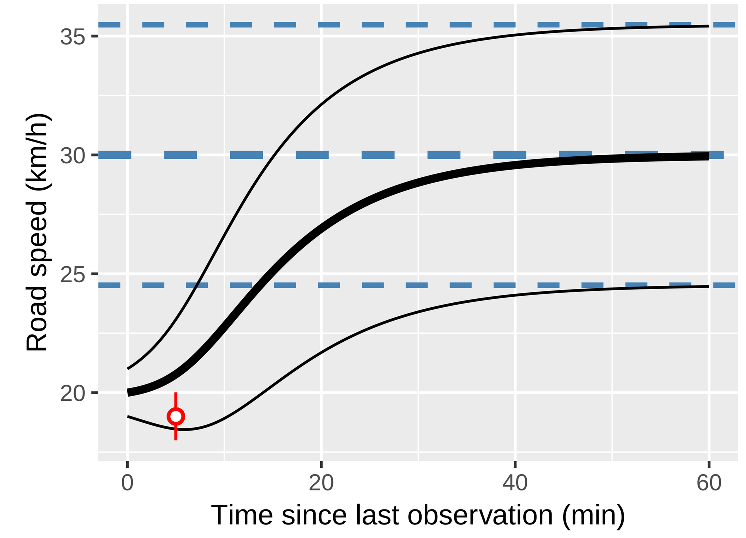


Figure 4: *After resampling, a posterior sample of trajectories is obtained (orange).*

4. Network state model

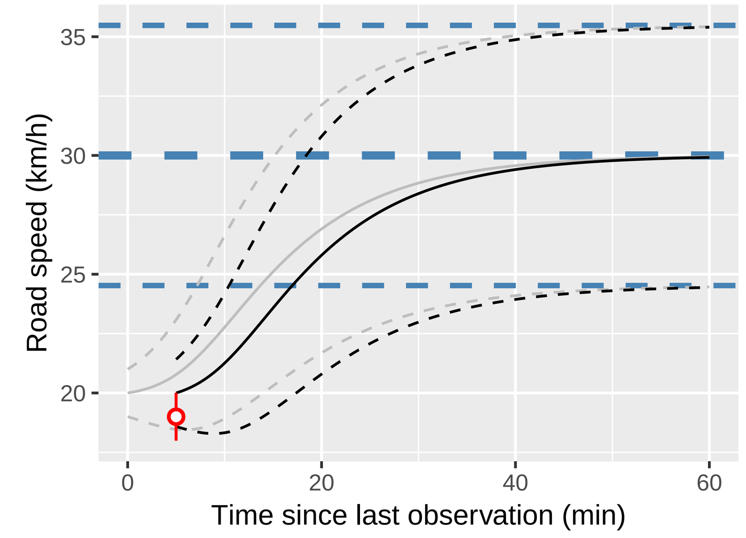
Predict step

- each segment j has state β_r^j (the speed of vehicles along the segment) at time t_r
- use historical data to determine the prior state $\mu_j(t)$ and $\psi_j(t)$, the mean and variance of speed at time t
- define transition function a such that the state converges to the prior
- use extended Kalman filter update (EKF) equations, and define system noise such that P_r^j, the uncertainty of travel time, converges to $\psi_j(t_r)$



Update step

- observations recieved when vehicles travel along segment
- measurement error calculated using combination of particle variance and between-vehicle variance (in cases where multiple vehicles are traveling a road at the same time)
- use EKF update equations to update state at time of observation
- repeat prediction step to obtain updated state forecasts



5. Predicting arrival time

- for each particle, simulate journey along remainder of route, **simulating speed** $v_t^j \sim N(\hat{\beta}_t^j, P_t^j)$ each time particle enters a new segment j
- simulate wait times at intersections and bus stops, and compute arrival time at each upcoming stop
- resulting ETA distribution can be conveyed to passengers
  - a point estimate
  - and/or a prediction interval, informing commuters how soon they need to be at their stop, but also prepare them for a possible wait

6. Conclusion and future work

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