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`

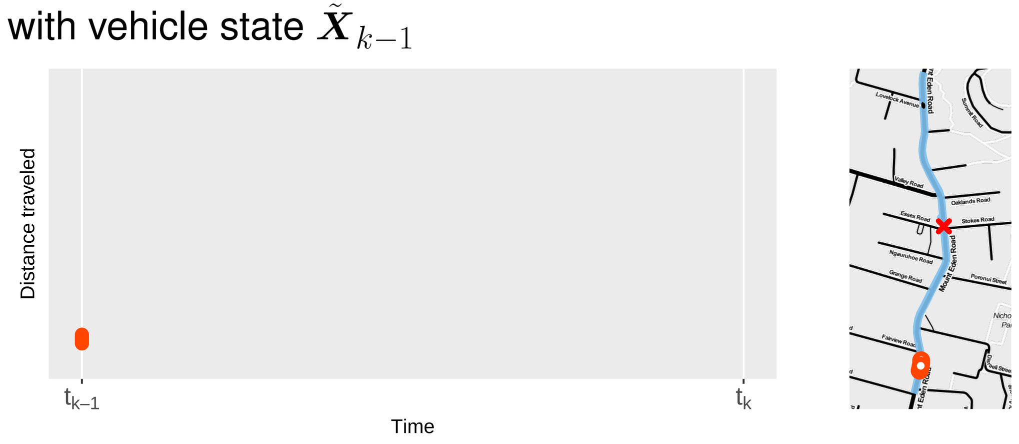
1. Introduction

- real-time prediction is hard, requires both current position and future travel times
- 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]
- 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. Vehicle state model

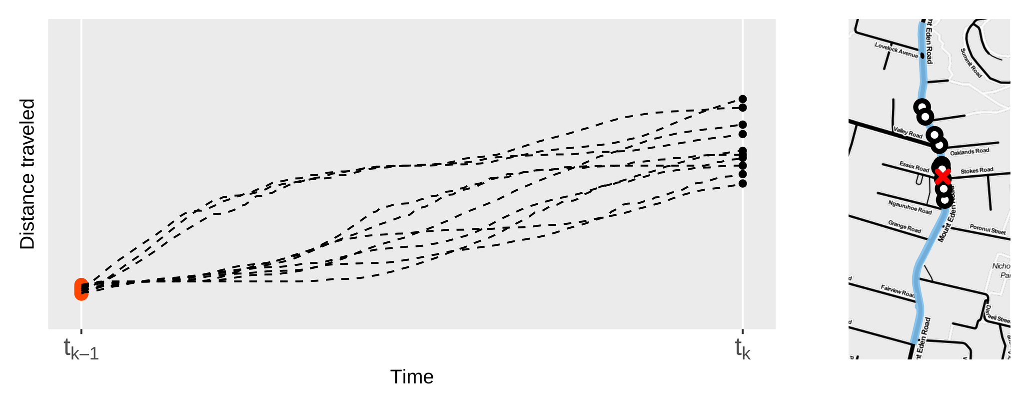
Goal: estimate vehicle state $\boldsymbol{X}_k$ (trip distance traveled and speed) from observed GPS positions, $\boldsymbol{Y}_k$, at time t_k

- sequential Bayesian methods well suited to real-time vehicle tracking
- the **particle filter** is a general, flexible estimation method using a sample of particles $\tilde{\boldsymbol{X}}_k = (\boldsymbol{X}_k^{(i)})_{i=1}^N$ to approximate state
- handles multimodality (e.g., when passing bus stops) and assymetry (e.g., bus cannot go backwards)
- involves two steps: *predict* future state, and *update* state using likelihood function
- measurement function $h : \mathbb{R} \mapsto \mathbb{R}^2$ calculates map (GPS) position of each particle based on distance traveled along shape
- start with vehicle state $\tilde{\boldsymbol{X}}_{k-1}$



- new observation recieved at time t_k
- transition function f predicts state of each particle based on previous state at time t_{k-1}, with system noise parameter Q_k

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



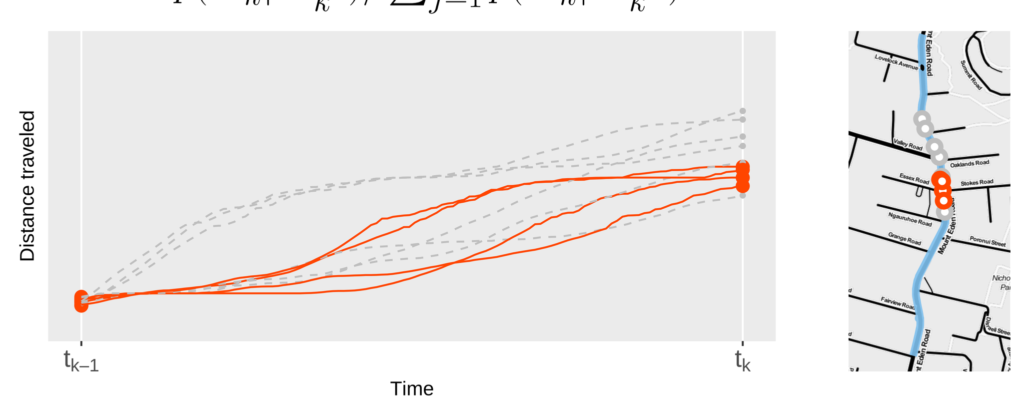
- assume observation $\boldsymbol{Y}_k$ is a noisy measurement of true position with error σ_y^2, and use geographic projection function g such that $dist(g(\boldsymbol{Y}_1), g(\boldsymbol{Y}_2))$ is the distance between the two points on the ground

$$g(\boldsymbol{Y}_k) \sim N\left(g(h(\boldsymbol{X}_k)), \begin{bmatrix}\sigma_y^2 & 0\\ 0 & \sigma_y^2\end{bmatrix}\right)$$

- $\delta_k^{(i)}$ is the distance between a particle position and observed position, which is the sum of two independent normal r.v.'s, giving the likelihood function

$$\left(\delta_k^{(i)}/\sigma_y^2\right) \sim \chi^2(2) \sim \text{Exp}(0.5)$$
$$p(\boldsymbol{Y}_k|\boldsymbol{X}_k^{(i)}) = 0.5e^{-\delta_k^{(i)}/2\sigma_y^2}$$

- update state by resampling particles with replacement, using likelihood weights $w^{(i)} = p(\boldsymbol{Y}_k|\boldsymbol{X}_k^{(i)}) / \sum_{j=1}^N p(\boldsymbol{Y}_k|\boldsymbol{X}_k^{(j)})$



- we now have estimate of vehicle speed at time t_k, which we can use in section 4

3. GTFS network construction

Goal: to represent each bus route as a sequence of physical road segments between intersections.

1. raw GTFS data provides one shape per route, represented as sequence of latitude/longitude coordinates
2. identify points of intersection between one or more routes using algorithm adapted from [6]
3. split shapes at intersection points to obtain shapes for each individual road segment
4. express each route as a sequence of road segments

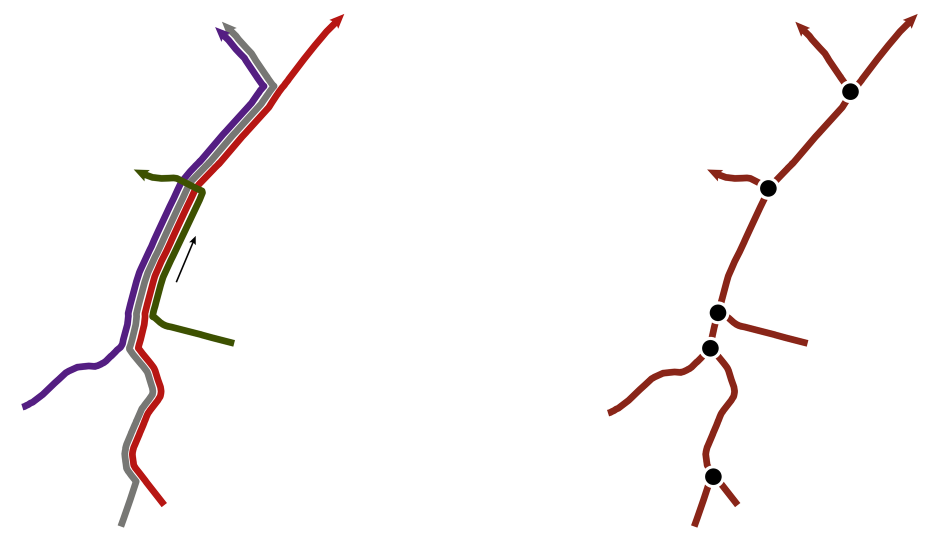


Figure 1: *An example transit netork produced from five routes.*

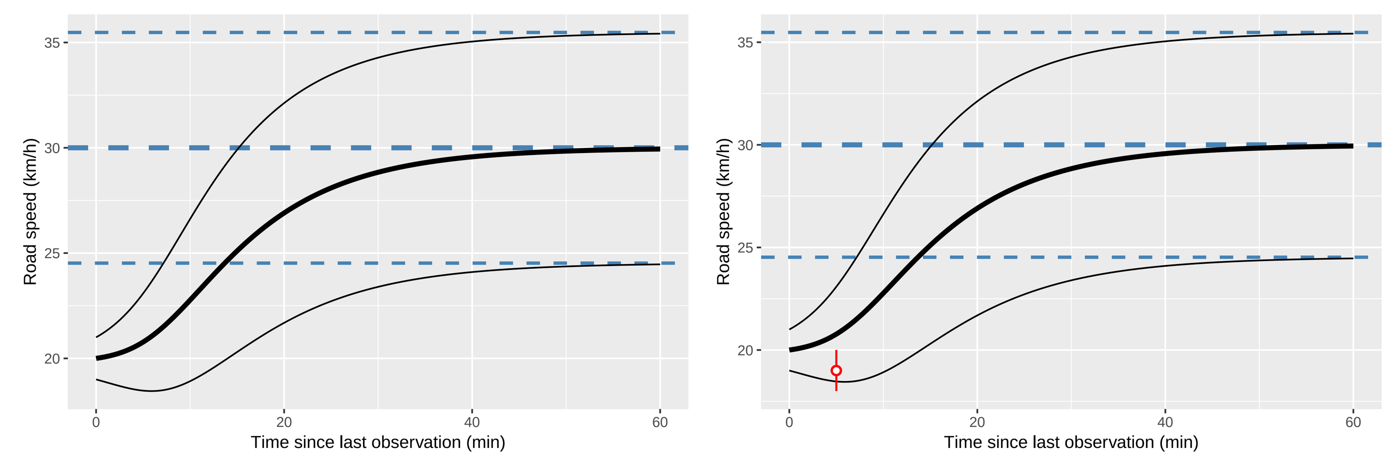
Currently being implemented in the `gtfsnetwork` package on GitHub: https://github.com/tmelliott/gtfsnetwork

4. Network state model

Goal: to model the travel time of transit vehicles along a road now, and in the near-future

- each segment j has state β_r^j (the travel time 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 travel time at time t
- define transition function a such that the state converges to the prior

$$\beta_r^j = a(\beta_{r-1}^j, P_{r-1}^j, \mu_j(t_r), \psi_j(t_r)) = \beta_{r-1}^j + \frac{\lambda P_{r-1}^j}{P_{r-1}^j + \psi_j(t_r)}(\mu_j(t_r) - \beta_{r-1}^j) + e_r^j, \quad e_r^j \sim N(0, S_j)$$



5. Arrival time prediction

6. Conclusion

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] 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