Tom’s awesome thesis on buses

Tom Elliott

2019

Contents

Abstract

This is the abstract.

Chapter 1

Introduction

This is the intro. Literature review, the problem, and the goals of this work (i.e., to make better predictions that don’t rely on the timetable).

Part I

Real-time bus models

Chapter 2

Literature review

This is a “review” of what’s been happening in the field of bus arrival-time prediction and modeling. From its routes in Kalman filtering, through fancy ANN and SVM models, to computer intensive particle filter models (not for real-time applications).

What’s missing: focus on improved arrival-time prediction, rather than an OR approach.

Chapter 3

GTFS data and route segmentation

Talk about the data itself: where it comes from, the important aspects we care about (trips/routes, shapes, stops, and stop times).

3.1 Real-time data

What's involved, frequency, and some of the quirks (at least in our Auckland Transport example).

3.2 Segmentation of routes

This is cool — by segmenting routes (at intersections) we remove dependency of speed/travel time on *route* and instead relate it to the physical road the vehicle is traveling along.

Chapter 4

Transit vehicle model

This chapter should be about the model itself, including traffic lights, dwell times, speed, etc. It's a recursive Bayesian model, which means the state at time k is a function of the state at time $k-1$.

4.1 Particle filter

About how it's actually implemented using a particle filter, and the reasons why we chose that.

Part II

Transit network

Chapter 5

Constructing a GTFS network

Here we give a detailed overview of what exactly the transit network is (in the previous chapter, we only alluded to it, and assumed segments where known).

The basic idea of splitting shapes at intersections, which we get manually, or could potentially obtain using shape-processing methods.

Some of the slight issues, e.g., missing intersections.

Chapter 6

Real-time network model

The *network state* is denoted by $\boldsymbol{\beta}$, a vector of length L, the number of segments in the network. Each β_ℓ represents the *mean travel time* of buses in segment ℓ. To denote time, we add a c subscript to the state, $\boldsymbol{\beta}_c$ and $\beta_{c\ell}$.

The *state estimate* will be denoted as $\hat{\boldsymbol{\beta}}$.

6.1 Predict step

The first step in the EKF process is to predict the state at time c. Here we wish to estimate $\hat{\boldsymbol{\beta}}_{c|c-1}$, the network state at time c, given the posterior state estimate at time $c-1$.

6.1.1 Transition function

Based on an extended Kalman filter, the transition function f defines the state at time c as a deterministic function of the state at time $c-1$. That is,

$$\hat{\boldsymbol{\beta}}_{c|c-1} = f(\hat{\boldsymbol{\beta}}_{c-1|c-1}). \tag{6.1}$$

The next step is to define f so that it meets the criteria for the EKF (it must be differentiable), and has the properties we desire: that the state will regress to the prior in the absense of any observations.

To bring the state towards the prior mean travel time for segment ℓ, ν_ℓ, we define the transition function for a single second, f, as

$$f(\beta_{\ell,c}, \nu_\ell, \lambda) = \beta_{\ell,c-1} + \lambda(\nu_\ell - \beta_{\ell,c-1}) \tag{6.2}$$

where $\lambda \in (0,1)$ is a tuning parameter that determines the rate at which the state converges. We can easily calculate the derivative of f, too, which is a requirement for the EKF.

$$f'(\beta_{\ell,c}, \nu_\ell, \lambda) = 1 - \lambda \tag{6.3}$$

Next we need to be able to define the a transition for Δ seconds, which by recursion of 6.2 is

$$f_n(\beta_{\ell,c-1}, \nu_\ell, \lambda) = (f^1 \circ \cdots \circ f^n)(\beta_{\ell,c=1}, \nu_\ell, \lambda), \quad \text{where } f^j \equiv f \quad \forall j. \tag{6.4}$$

The derivative of 6.4 is obtained used the chain rule, $f(g(x))' = f'(g(x))g'(x)$. We will use $f(\beta) \equiv f(\beta_{\ell,c-1}, \nu_\ell, \lambda)$ to simplify notation (the other parameters are constant).

$$\begin{aligned}
\mathrm{F}_n = f_n'(\beta) &= f'\left((f^1 \circ \cdots \circ f^{n-1})(\beta)\right)(f^1 \circ \cdots \circ f^{n-1})'(\beta) \\
&= f'\left(f_{n-1}(\beta)\right)\mathrm{F}_{n-1} \\
&= f'\left(f_{n-1}(\beta)\right) f'\left(f_{n-2}(\beta)\right) \cdots f'\left(f_1(\beta)\right) f'(\beta) \\
&= f'(\beta) \prod_{i=1}^{n-1} f'\left(f_i(\beta)\right)
\end{aligned}$$

Substituting 6.3, we get

$$\mathrm{F}_n = (1-\lambda) \prod_{i=1}^{n-1} (1-\lambda) = (1-\lambda)^n \tag{6.5}$$

Next we need to ensure the entire state, and not just the mean, converges to the prior, i.e., the state variance $P_{\ell,c}$ converges to the prior variance, $\xi_\ell = \xi_\ell$.

The system noise, Q, is used to predict the state variance,

$$P_{\ell,c|c-1} = F_n^2 P_{\ell,c-1|c-1} + Q \tag{6.6}$$

At convergence, we want $P_{\ell,c|c-1} = P_{\ell,c-1|c-1} = \xi_\ell$. Substituting into 6.6 and solving for Q:

$$\begin{aligned}
\xi_\ell &= F_n^2 \xi_\ell + Q \\
&= (1-\lambda)^{2n} \xi_\ell + Q \\
Q &= \xi_\ell (1 - (1-\lambda)^{2n}) \qquad (6.7)
\end{aligned}$$

Now that the state will converge to the prior, the last thing we have to do is determine how fast it does so. This is controlled by λ; however, we want the rate of convergence to be slow after an observation, and speed up the longer we go without seeing any buses. That is, we want to use a ratio of P_ℓ and ξ_ℓ,

$$\lambda = \phi \frac{P_{\ell,c-1|c-1}}{P_{\ell,c-1|c-1} + \xi_\ell} \tag{6.8}$$

where ϕ is a tuning parameter.

6.2 Update step

Having predicted the state at time t_c, we can now update our estimate based on the observed data, $\mathbf{B}_c$, which consists of the combined travel times of all K_m vehicles completing travel along the segment during the current iteration. For each vehicle, we can obtain the mean and uncertainty of travel time from the particle filter, $b_{v,\ell}$ and $e_{v\ell}$, respectively. These can then be combined to obtain a single observation for the EKF, along with an estimate to use as measurement error:

$$B_{\ell,c} = \frac{1}{K_m} \sum_{v=1}^{K_m} b_{v,\ell} \tag{6.9}$$

$$E_{\ell,c} = \frac{1}{K_m} \sum_{v=1}^{K_m} \left(b_{v,\ell}^2 + e_{v,\ell}^2 \right) - B_{\ell,c} \tag{6.10}$$

Chapter 7

Historical data based priors

How we use Bayesian heirarchical models to estimate the typical speed along roads in the network, based on historical data.