

Desafío TEA

Concurso para desarrolladores independientes
Octubre 2018

GIT: <https://github.com/cdotta/teayuda/>

Gustavo Armagno
Cristian Dotta
Mauricio Gonzalez
Javier Schandy

CI: 3.895.619-0
CI: 4.737.975-1
CI: 4.632.731-5
CI: 4.551.376-7

Introducción

El objetivo del Desafío TEA consiste en el diseño y la implementación de un algoritmo capaz de devolver Tiempos Estimados de Arribo para buses que den servicio a líneas incluidas en el STM de Montevideo.

Dicho algoritmo utiliza la Plataforma de Ciudad Inteligente de Montevideo (basada en Fiware) para obtener información en tiempo real relativa a la ubicación y el servicio de cada uno de los buses. Además, obtiene información relativa a la ubicación de cada una de las paradas de las diferentes líneas.

Características del problema

El transporte público montevideano se compone de una red de paradas conectadas por varias líneas que son operadas por diferentes compañías. El sistema de transporte es esencialmente descentralizado, y la mayoría de las paradas forman parte del recorrido de más de una línea.

Por otra parte, la aplicación limitada de sendas preferenciales para ómnibus hace que estos tengan una interacción elevada con el resto de los vehículos, lo que aumenta la incertidumbre en los tiempos estimados de llegada.

Modelado del problema

El modelado del sistema se divide en dos partes. En primer lugar, se plantea una representación que divide el problema en múltiples subsistemas conformados por los enlaces entre cada par de paradas. A continuación, se plantea un modelo que permite predecir el tiempo necesario para recorrer la ruta entre dos paradas.

De aquí en adelante se utilizarán los siguientes términos

- Coche: Bus en servicio.
- Parada: Lugar donde un coche permite el ascenso y descenso de pasajeros.
- Tramo: Trayecto entre paradas
- Línea: Conjunto de tramos que recorre un coche durante su servicio.

Por otra parte, se distinguirán las medidas de tiempo estimado de arribo t_k de sus estimados \hat{t}_k mediante el acento "^".

Modelado básico

Si bien existen numerosos trabajos publicados orientados a la estimación de TEA para un bus, estos no suelen atacar el problema para varias líneas con paradas en común, y suelen modelar buses que utilizan sendas exclusivas. [1][2][3][4]

Frente a este escenario, proponemos un modelo inspirado en el utilizado en [1]. Este trabajo resuelve el problema de TEA para tres líneas que comparten segmentos de sus rutas. Las rutas son segmentadas en los diferentes tramos entre paradas, y cada segmento es modelado independientemente.

En nuestro caso, el entramado de paradas es considerablemente más complejo que en el caso anterior, por lo que lo modelamos mediante un grafo en el que cada parada es un nodo, mientras que las diferentes líneas se descomponen en arcos que los conectan.

Sean

- A, B, C, \dots las paradas del entramado
- $1, 2, 3, \dots$ las líneas que conectan las paradas
- x, y, z, \dots los diferentes coches que sirven las líneas

Podemos ejemplificar nuestro modelo según la Figura 1.

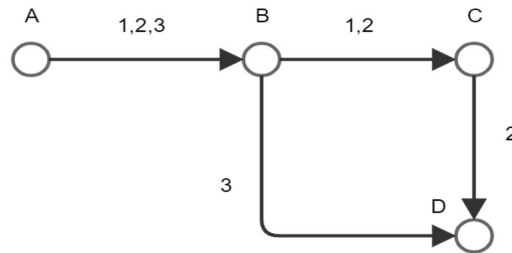


Figura 1. Ejemplo simplificado de grafo que modela el conjunto de paradas

En este ejemplo, las líneas 1, 2 y 3 conectan las paradas A y B (tramo \overline{AB}), las líneas 1 y 2 conectan el tramo \overline{BC} , la línea 2 conecta el tramo \overline{CD} y la línea 3 conecta el tramo \overline{BD} .

De manera análoga a [4], modelaremos cada uno de los tramos bajo la hipótesis de que el tiempo que necesita un bus para recorrer el tramo entre dos paradas no depende de la línea que está sirviendo. Esto significa que dado un tramo \overline{XY} , dos líneas n, m y los tiempos estimados para recorrer el tramo para un bus de cada una de las líneas, $\hat{t}_{n,XY}, \hat{t}_{m,XY}$; se cumple que $\hat{t}_{n,XY} = \hat{t}_{m,XY}$.

Problema de estimación de TEA

Sobre este modelo, el problema de “estimar el tiempo necesario para que el siguiente bus de la línea n llegue a la parada Y ” se resume a

- Encontrar X , la última parada que visitó el coche de la línea n más cercano a Y , recorriendo el grafo hacia atrás.
- Estimar \hat{t}_{XY} , el tiempo necesario para recorrer las paradas X a la Y

La estimación del tiempo necesario para realizar el recorrido de X a la Y se obtiene como

$$\hat{t}_{XY} = \sum_{n=X}^Y \hat{t}_{n,n+1}$$

O lo que es lo mismo, equivale a la suma de las estimaciones de tiempo necesario para recorrer todos los tramos intermedios entre X y Y .

Obtención de estimadores para cada uno de los tramos

En base a lo discutido en la sección anterior, el problema se reduce a obtener buenos estimadores para cada uno de los tramos. Para este fin se propone la estructura de filtros que se muestra en la Figura 2.

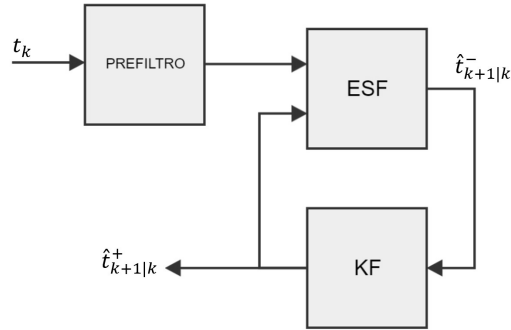


Figura 2. Cadena de filtros utilizados

Para simplificar la notación estudiaremos un único tramo \overline{XY} , cuya identidad no se incluye en el nombre de ninguna variable o parámetro.

En la figura anterior

- t_k es una medida del tiempo que le llevó a un bus recorrer tramo, sin considerar su línea.
- $\hat{t}_{k+1|k}^-$ es el *estimado a priori* del tiempo que necesitará el próximo bus para recorrer el tramo
- $\hat{t}_{k+1|k}^+$ es el *estimado a posteriori* del tiempo que necesitará el próximo bus para recorrer el tramo, y en definitiva es el estimado que se considera válido

Por otra parte, los bloques son

- ESF: Filtro de suavizado exponencial. Es el modelo escogido para obtener un estimado del tiempo que necesitará el próximo bus para recorrer el tramo basado en los tiempos anteriores
- KF: Filtro de Kalman. Filtro utilizado fusionar el estimado anterior con datos provenientes de otras fuentes, además de filtrar diferentes fuentes de ruido.
- Prefiltro: Se trata de un bloque lógico que elimina algunas medidas consideradas erróneas, por criterios que serán discutidos más adelante

Funcionamiento del filtro de suavizado exponencial (ESF) como predictor de TEA

Dada una medida de tiempo t_k y el estimado que se había realizado para ese tiempo, \hat{t}_k , el siguiente estimado \hat{t}_{k+1} se define como:

$$\hat{t}_{k+1} = \hat{t}_k + \alpha(t_k - \hat{t}_k)$$

Es decir, una nueva estimación se compone de la estimación anterior y su error, multiplicado por un factor α . Es interesante reordenar los términos de la siguiente manera.

$$\hat{t}_{k+1} = \alpha t_k + (1 - \alpha)\hat{t}_k$$

Si sustituimos la expresión de \hat{t}_k en \hat{t}_{k+1}

$$\hat{t}_{k+1} = \alpha t_k + (1 - \alpha)(\alpha t_{k-1} + (1 - \alpha)\hat{t}_{k-1})$$

Sustituyendo recursivamente la misma expresión y reordenando, llegamos a la expresión

$$\hat{t}_{k+1} = \alpha t_k + \alpha(1 - \alpha)t_{k-1} + \alpha(1 - \alpha)^2 t_{k-2} + \alpha(1 - \alpha)^3 t_{k-3} + \dots + \alpha(1 - \alpha)^k t_0$$

$$\hat{t}_{k+1} = \alpha \sum_{i=0}^{k-1} (1 - \alpha)^{k-1-i} t_i$$

Lo que demuestra que el estimado del filtro es un promedio ponderado exponencialmente de toda la historia de los buses que han pasado por este tramo.

Esta cualidad resulta particularmente interesante para nuestra aplicación, ya que

- Los tramos con una gran cantidad de recorridos por unidad de tiempo (es decir, los más céntricos y por ende con más tránsito) asignarán un peso menor a medidas antiguas, y podrán reaccionar a cambios de velocidad en el tránsito con más rapidez.
- Los tramos periféricos, recorridos por pocos buses por unidad de tiempo darán un peso mayor a los valores antiguos, pero es esperable que los estimados para estos tramos se mantengan aproximadamente constantes durante el día.

Según nuestra evaluación experimental, elegimos usar un $\alpha = 0.3$. En [2] se sugiere un $\alpha = 0.5$

Integración del ESF con el filtro de Kalman

El filtro de Kalman es un filtro utilizado básicamente con dos propósitos

- Es el filtro de *fusión sensorial* por excelencia. Esto quiere decir que permite combinar las medidas provenientes de diferentes fuentes, devolviendo un estimado único a partir de ellas. Este estimado unificado considera el grado de precisión (es decir, la varianza) de cada fuente.
- Adicionalmente permite modelar el impacto de las diferentes fuentes de ruido que afectan nuestro sistema.

Este filtro se alimenta de las estimaciones realizadas para el instante k por el ESF (que a partir de ahora llamaremos *estimación a priori*, y notaremos \hat{t}_{k+1}^-) y de una medida real del estimado anterior, que notaremos z_k para entregarnos una nueva estimación, llamada *estimación a posteriori* para el instante $k + 1$, que es representada mediante \hat{t}_{k+1}^+ .

Como podemos intuir, el modelo ESF está lejos de ser un modelo exacto del sistema que estamos estudiando (cualquiera de los tramos). Esta imprecisión puede ser modelada mediante un ruido aditivo, llamado *ruido de proceso*.

Como ya definimos, la expresión del ESF es

$$\hat{t}_{k+1} = (1 - \alpha)\hat{t}_k + \alpha t_k$$

Kalman incorpora a esta ecuación un nuevo término w_k que representa el ruido de proceso, consistente en una variable aleatoria con distribución gaussiana de media nula y varianza Q_k . La nueva expresión es

$$\hat{t}_{k+1} = (1 - \alpha)\hat{t}_k + \alpha t_k + n_k$$

Además de modelar el ruido generado por efectos no representados en el modelo, el filtro de Kalman permite integrar medidas provenientes de otras fuentes. Estas medidas externas, que llamaremos z_k , tampoco son completamente confiables ya que fueron tomadas por sensores con ruido asociado.

Sea t_k el tiempo real que requirió el último bus en recorrer el tramo, la expresión de z_k es

$$z_k = H_k \cdot t_k + v_k$$

Con v_k una variable aleatoria gaussiana con distribución gaussiana de media nula y varianza R_k

Las medidas z_k pueden provenir de varias fuentes. Dos de las propuestas que consideramos más interesantes son:

- De manera análoga a lo realizado en [2], integrar medidas de demora entre paradas provenientes de Orion, el mismo día a la misma hora de la semana anterior.
- Usando la API de Google Maps, realizar una consulta para obtener una medida de la demora entre dos paradas para el instante actual

En nuestro caso no podemos usar ninguno de los dos métodos ya que no tenemos datos históricos de Orion, y no podemos realizar consultas a Google Maps para fechas pasadas. Por estas razones, y solamente a efectos de la demo del concurso elegimos usar $z_k = \hat{t}_{k+1}^-$, lo que debería ser corregido en una aplicación real.

Enunciado del problema de Filtrado de Kalman¹

Dado un sistema de la forma

$$\begin{cases} x_{k+1} = F_k x_k + G_k w_k \\ z_k = H'_k x_k + v_k \end{cases}$$

Con $\{w_k\}$, $\{v_k\}$ variables aleatorias gaussianas, independientes entre si y con distribuciones $N(0, Q_k)$ y $N(0, R_k)$, la estimación \hat{x}_{k+1} que minimiza el error cuadrático medio es

$$\hat{x}_{k+1} = [F_k - K_k H'_k] \hat{x}_k + K_k z_k$$

Con

$$K_k = F_k \Sigma_{k|k-1} H'_k [H'_k \Sigma_{k|k-1} H_k + R_k]^{-1}$$

$$\Sigma_{k+1|k} = F_k \left[\Sigma_{k|k-1} - \Sigma_{k|k-1} H_k (H'_k \Sigma_{k|k-1} H_k + R_k)^{-1} H'_k \Sigma_{k|k-1} \right] F'_k + G_k Q_k G'_k$$

Afortunadamente estas ecuaciones matriciales se vuelven bastante sencillas en nuestro caso unidimensional. Las ecuaciones usadas en nuestro caso fueron:

- $\hat{t}_{k+1}^- = (1 - \alpha) \hat{t}_k^+ + \alpha t_k + n_k$
- $\Sigma_{k|k} = (1 - \alpha)^2 \Sigma_{k|k} + Q_k$
- $K_{k+1} = \Sigma_{k|k} \cdot \frac{1}{\Sigma_{k|k} + R}$
- $\hat{t}_{k+1}^+ = \hat{t}_{k+1}^- + K_{k+1} (z_{k+1} - \hat{t}_{k+1}^-)$
- $\Sigma_{k+1|k} = (1 - K_{k+1}) \Sigma_{k|k}$

Prefiltro

El prefiltro cumple la función de evitar que ingresen valores al sistema de filtros que son claramente incorrectos. Los valores de demora medidos se rechazan por tres criterios.

¹ Versión simplificada del enunciado. Una versión más completa, con su correspondiente demostración puede consultarse en [5]

- Si la duración del tramo es menor a 15s
- Si la duración del tramo es mayor a 600s
- Si la información del bus no fue refrescada por más de 90s

Mejoras adicionales

Proyección de tiempo estimado

Es muy probable que, al recibir una consulta sobre el TEA para una determinada línea, el bus más cercano no se encuentre en una parada, sino que se encuentre realizando el trayecto entre dos puntos por lo que no tendría sentido sumar el TEA para el tramo completo.

Por otra parte, como no contamos con información sobre el recorrido real de un bus, sino solamente con la ubicación de sus paradas, no podemos calcular un “porcentaje de realización” del recorrido del tramo.

Para resolver este problema, se utiliza la proyección de la posición del bus sobre la línea que une dos paradas, tal como se muestra en la Figura 3.

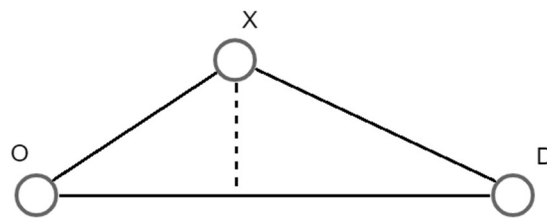


Figura 3. Algoritmo de proyecciones para recorridos en curso

Si bien no es una aproximación exacta, se considera satisfactoria ya que

- Solamente se usa en el tramo que está siendo recorrido en ese instante
- Las paradas de Montevideo no suelen ser demasiado distantes entre sí
- Los buses tienden a tener paradas antes o después de doblar una esquina

En la Figura 4 se muestran dos casos (405, 183) en los que estos problemas son particularmente significativos, pero aun así no parecen tener un gran impacto.

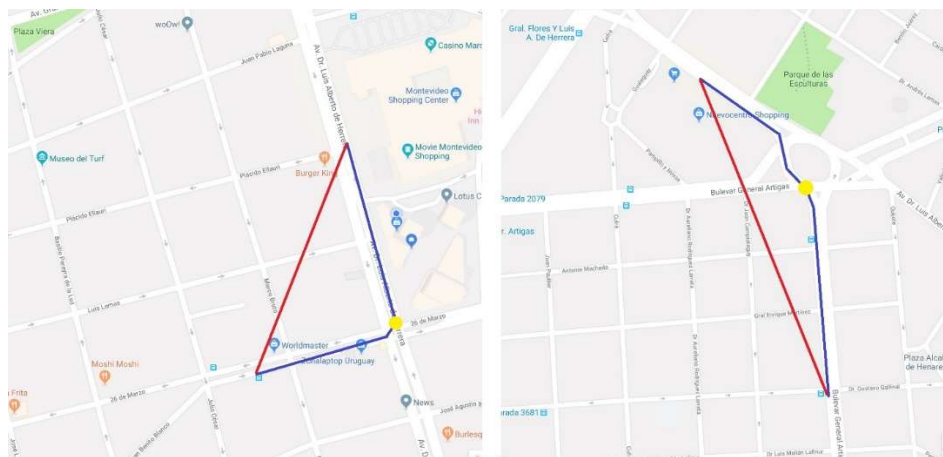


Figura 4. Ejemplos de paradas en las que la proyección introduce un error.

Dadas las coordenadas de una parada de origen (O) y otra de destino (D), y las coordenadas actuales del bus (X), se define el parámetro β como

$$\beta = \frac{\text{proyección de } \overline{XD} \text{ en } \overline{OD}}{\overline{OD}}$$

Aplicando trigonometría,

$$\beta = \frac{\sqrt{\overline{XD}^2 - h_x^2}}{\overline{OD}}$$

Con $h_x = \frac{2}{\overline{OD}} \sqrt{s(s - \overline{OX})(s - \overline{XD})(s - \overline{OD})}$ y $s = \frac{\overline{OX} + \overline{XD} + \overline{OD}}{2}$

El parámetro β permite obtener una métrica del grado de avance que tuvo el bus entre los puntos de origen y destino. El TEA del bus a D desde el punto X ($\hat{t}_{X,D}$) se calcula como

$$\hat{t}_{X,D} = \beta \hat{t}_{O,D}$$

Con el objetivo de filtrar casos patológicos se impone que $0 < \beta < 1$

Implementación

La solución fue implementada en NodeJS V8, siguiendo la arquitectura que se muestra en la Figura 5

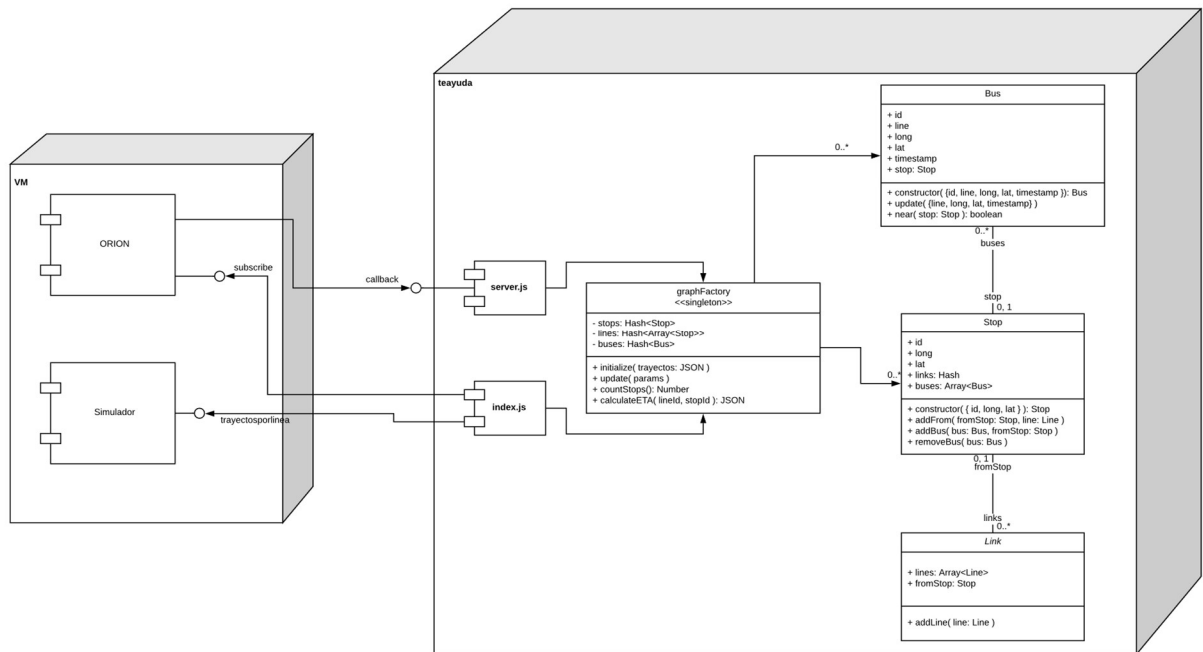


Figura 5. Arquitectura de la solución implementada

La Figura 6 muestra la representación de alto nivel de las entidades modeladas (Parada, Bus) y sus relaciones.

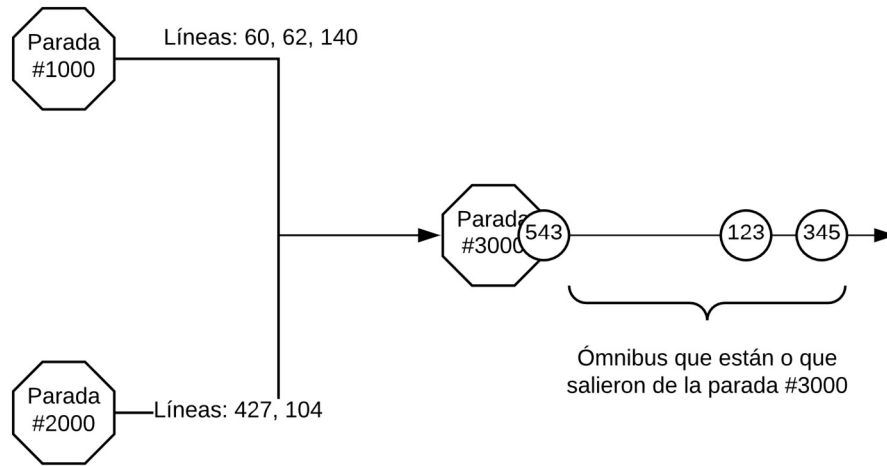


Figura 6. Representación de entidades

La Figura 7 muestra las estructuras de datos utilizadas para representar internamente estas entidades y sus relaciones.

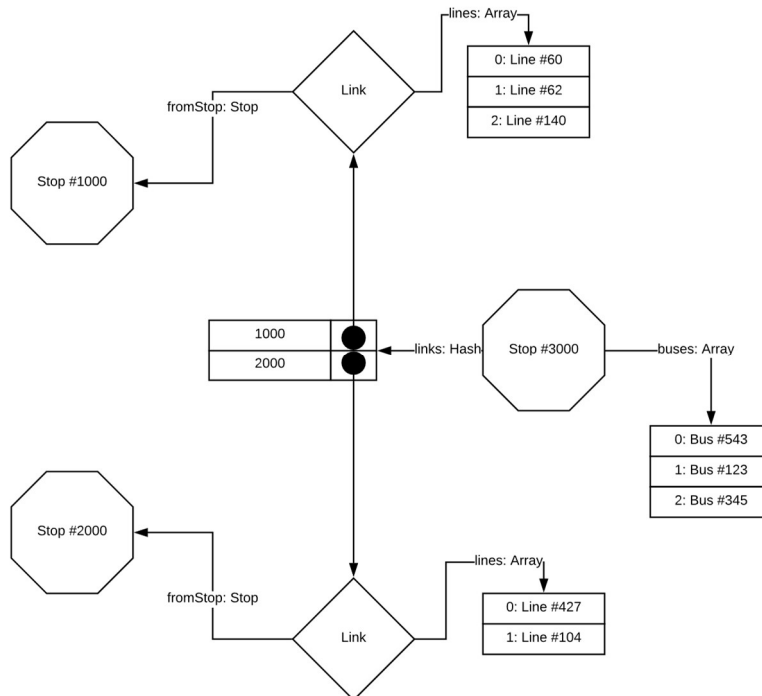


Figura 7. Estructuras de datos

La figura 8 muestra las estructuras auxiliares utilizadas para facilitar las operaciones de actualización y recuperación de datos.

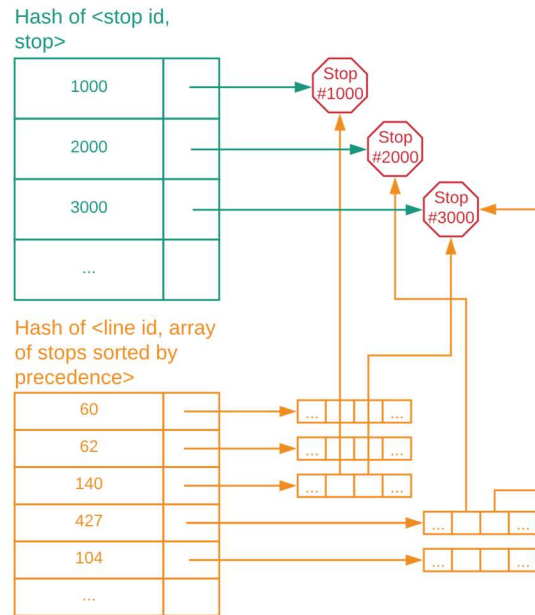


Figura 8. Estructuras auxiliares

Características del modelo

- Estimadores fácilmente modificables e integrables con otras fuentes de información

El filtro de Kalman es ampliamente reconocido como **una de las mejores técnicas para fusionar fuentes de información**. En el caso de querer escalar el servicio a una aplicación real, **sería sumamente sencillo integrar fuentes de datos** basadas en tráfico como Google Maps o Bing Maps.

La integración de fuentes de información a través del filtro de Kalman garantiza que la varianza del resultado es menor que la varianza de cualquiera de las entradas.

- No requiere grandes volúmenes de datos históricos para funcionar

Dado que el enfoque utilizado se basa en filtros adaptivos en lugar de en machine learning, **nuestro sistema es capaz de devolver predicciones utilizando solamente registros de algunos minutos**.

- Resistente a ruido en los datos

Nuestro sistema estima el TEA en base a transiciones entre paradas, por lo que es **sumamente robusto a ruido en la posición** de los buses. Otros sistemas que basen las predicciones en la cinemática de los puntos pueden desestabilizarse enormemente si, por ejemplo, se reciben coordenadas $(0, 0)$. En nuestro caso es prácticamente imposible que este fenómeno tenga impacto en el TEA estimado, ya que para que esto ocurra el punto afectado debería caer en una parada posterior de la misma línea.

- Resistente a pérdidas de datos

Nuestro modelo es capaz de detectar la pérdida de reportes de un bus por largos períodos, eliminándolo de las predicciones. Cuando vuelve a reportarse, se vuelve a integrar su posición en las predicciones.

- Prevé la posibilidad de que un bus cambie de línea

El sistema se adapta muy naturalmente a cambios en los datos provenientes de Orion. Un cambio en la línea que sirve un bus desencadenaría la creación de un nuevo bus, sin generar ningún inconveniente.

Bibliografía

- [1] Kumar, S. V., Chaitanya Dogiparthi, K., Vanajakshi, L., & Subramanian, S. C. (2017). Integration of exponential smoothing with state space formulation for bus travel time and arrival time prediction. *Transport*, 32(4), 358-367.
- [2] Yu, B., Lam, W. H., & Tam, M. L. (2011). Bus arrival time prediction at bus stop with multiple routes. *Transportation Research Part C: Emerging Technologies*, 19(6), 1157-1170.
- [3] Xumei, C., Huibo, G., & Wang, J. (2012). BRT vehicle travel time prediction based on SVM and Kalman filter. *Journal of transportation systems engineering and information technology*, 12(4), 29-34.
- [4] Hua, X., Wang, W., Wang, Y., & Ren, M. (2018). Bus arrival time prediction using mixed multi-route arrival time data at previous stop. *Transport*, 33(2), 543-554.
- [5] Anderson, B. D., & Moore, J. B. (1979). Optimal filtering. *Englewood Cliffs*, 21, 22-95.