Public Transport (2021) 13:59–80
https://doi.org/10.1007/s12469-020-00255-9

ORIGINAL RESEARCH

Estimation and prediction of dynamic matrix travel on a public transport corridor using historical data and real-time information

Felipe Zúñiga[1] · Juan Carlos Muñoz[1,2] · Ricardo Giesen[1,2]

Accepted: 9 September 2020 / Published online: 2 November 2020

Abstract

In this paper a new methodology to estimate/update and forecast dynamic real time origin–destination travel matrices (OD) for a public transport corridor is presented. The main objective is to use available historical data, and combine it with online information regarding the entry and exit of each particular user (e.g. through the fare system, FS), to make predictions and updates for the OD matrices. The proposed methodology consists of two parts: (1) an estimation algorithm for OD matrices of public transport (EODPT), and (2) a prediction algorithm (PODPT) based on artificial neural networks (ANNs). The EODPT is based on a model that incorporates the travel time distribution between OD pairs and the modeling of the travel destination choice as a multinomial distribution, which is updated using a Bayesian approach with new available information. This approach makes it possible to correct the estimates of both the current OD interval matrices and of preceding intervals. The proposed approach was tested using actual demand data for the Metro of Valparaiso corridor in Chile (Merval), and simulated travel information in the corridor. The results are compared favorably with a static approach and can support the use of this methodology in real applications. The execution times obtained in the test cases do not exceed 10 s.

Keywords OD matrices estimation · OD matrices prediction · Real-time

✉ Juan Carlos Muñoz
jcm@ing.puc.cl

Felipe Zúñiga
fazuniga@uc.cl

Ricardo Giesen
giesen@ing.puc.cl

[1] Department of Transport Engineering and Logistics, Pontifical Catholic University of Chile, Vicuña Mackenna 4860, Macul, Casilla 306, Correo 22, Santiago, Chile

[2] Centro de Desarrollo Urbano Sustentable, CEDEUS, Santiago, Chile

1 Introduction

The operation of public transport systems is characterized by the variability in both travel times between successive stops and passenger demand in each of them. This variability tends to produce vehicle bunching, which ends up deteriorating waiting times, service reliability and comfort measures.

In order to avoid these negative impacts on the level of service, certain control measures can be implemented in a way to regularize the time intervals between vehicles. The results reported by Delgado et al. (2009, 2012), show that schemes which combine such control measures delivered encouraging results, offering even levels of service for users and significant improvements over those obtained with each measure individually.

The availability of new technologies offers extremely advantageous opportunities in the creation and implementation of systems that control and regulate the evolution of public transport systems. Many systems currently use technologies of automatic passenger counting and automatic fare collection, in which the fare depends on the origin and destination of the trip of the user. In such systems users are identified (for example, through the validation of their payment card) at the beginning and end stops of their trip. Thus, this system allows to know the origin and destination stop, the departure and arrival times (and travel time), and the type of user. This system with validation both in the origin and destination stop will be called VOD.

Traditionally, control schemes deliver satisfactory results in regular demand scenarios, outlined in an average historical OD matrix. However, this information is not always sufficient, particularly when disruptions or unexpected events occur in the system, whose demand patterns are not captured in the historical matrix. Thus, a mechanism that allows updating the historical information online using data being obtained in real time, provides a powerful tool that would meet this need.

In this scenario, using the aforementioned technology systems would provide updated and higher quality information about trips made in the system which should allow predicting short-term future demand (e.g. next 15–30 min) in a more precise way. For this reason, the motivation for this research is to exploit such systems, in order to generate more accurate estimates and predictions of the present and future state of the system—dynamic OD matrices[1]—that would increase the benefits provided by the control schemes.

This process of estimation and prediction of trip matrices should be fast enough for real-time implementation. This speed contrasts with the high cost and long duration required by traditional processes for dynamic OD matrices. The OD matrix estimation has been extensively discussed in the literature (see, for example, Ashok and Ben-Akiva 1993; Sherali and Park 2001; Bierlaire and Crittin 2004; Zhou and Mahmassani 2007; Carrese et al. 2017; Krishnakumari et al. 2020) works mainly focused on private transport. As for public transport, not only are there less authors who have worked on the subject, see, e.g. Nguyen and Pallotino (1986), Nguyen et al.

[1] The term "dynamic" refers to the update of the OD matrix in successive intervals/times in a day.

(1988), Wong and Tong (1998), Li (2008), Rahman et al. (2016) Toqué et al. (2016), and Jenelius (2019), but only recently Zhang et al. (2017) considers an online estimation about entry and exit of passengers to predict OD matrices. In our case, since we have online information about the station in which passengers board and alight we can use that information to estimate and update the forecast in real-time.

Thus, the aim of this paper is to present a methodology for estimation/update and forecast in real time of dynamic OD matrices for short intervals of time on a public transport corridor equipped with VOD.

The new methodology considers the estimation/update and prediction of dynamic OD matrices from a parametric approach. This means that an estimate of each of the cells of the OD matrices involved are not performed, but instead a model of travel destination choice is used, and the parameters associated with that model are estimated and updated.

For each origin stop, the proposed model uses a probability distribution of travel destination choice for each time interval. These probability distributions are calculated using the historical information of trips made in the corridor, information also used to extract the empirical travel time distributions between different OD pairs, throughout the day.

The process of estimation/update and prediction of OD matrices requires discretizing time in intervals of equal length. By working with these intervals, it is clear that there will be trips that begin within an interval (which will conform the OD matrix of that interval) and end in a future interval. Thus, the distribution of the destinations of trips beginning within a time interval will be modified in subsequent intervals, through the process of estimation/update as online information becomes available.

The methodology presented in this paper is based, then, mainly on two steps: first, a estimation/update phase, and second, a prediction phase. The first is responsible for performing the successive estimates of OD matrices of a public transport corridor (EODPT) associated to past time intervals, and update data needed to make forecasts of future states of the system. Meanwhile, the latter makes predictions of OD matrices of the public transport corridor (PODPT) associated with future time intervals.

These two algorithms are closely related. The first one will use known information, such as historical data and information collected from the fare system throughout the day, to produce/update estimates of OD matrices that are fed into the second algorithm. This second algorithm forecasts future OD matrices of the system, and does not provide any information to the estimation step.

The EODPT algorithm uses data related to how users enter and exit the system, which will also give information of when users are expected to get out of the system, according to observed time distributions. Notice that these estimates about when and where users leave the system has no relation to future OD matrices, since they only explain how current estimates of OD matrices will be distributed in time and space.

On the other hand, the PODPT algorithm also uses historical information and is also fed with an updated estimate about OD matrices of the latest time intervals, which are obtained from the EODPT algorithm. This algorithm provides forecasts for future time intervals, where no real information about those time intervals is available at the instant of generating the forecast.

Neural network techniques have shown great effectiveness when working with data of different natures, for example, demand profiles in public transport corridors and calendar information (day of week, holidays, etc.), and additionally have low times for delivering results. Based on these two attributes, they were chosen as an alternative to form the predictive engine of this work. It should be noted, however, that the main contribution of this work lies in the modeling used for updating OD matrices, whereby the travel destination of a user is the result of a multinomial process. The developed Bayesian algorithm has closed and very simple mathematical formulas, which result in very low time incurred to deliver results, enabling the proposed methodology to be used in real-time applications.

This article is organized as follows. In Sect. 2, there is a presentation and description of the problem, and a description of how the generic corridor under analysis works. Section 3 introduces the theoretical framework considered in this study, presenting the models to address the problem and the techniques used in the solution. Section 4 formally presents the formulation of the proposed methodology. In Sect. 5 the results of the application of the proposed methodology to a case study with real data are presented and finally Sect. 6 summarizes the main conclusions.

2 Presentation and description of the problem

2.1 Problem formulation

An analysis period T is considered, divided into equal intervals $k = 1, \ldots, K$, each of length Δ (e.g. 15 min). The network considered is a two-way public transport corridor, serving N stops. Thus, there are $n_{OD} = N \cdot (N - 1)$ OD pairs in the network.

For each interval k two types of information are available: (1) historical background, corresponding to an OD matrix that reflects the expected number of trips between each OD pair starting within that interval, as well as data of trips made during this interval in days before, and (2) the information captured in real time from users getting in and coming out of the system.

It is assumed that the current time corresponds to the end of interval $h = \left[h^{-}, h^{+}\right)$, where h^{-} and h^{+} define the beginning and end of that interval.

The main objective is to develop a methodology to generate estimation/update OD matrices observed in the corridor intervals $k \leq h$, and additionally forecast the OD matrices for $h < k \leq K$, i.e. future time intervals. The notation and general definitions of the problem are summarized in Zúñiga (2011).

2.2 Description of the analyzed base corridor and its fare system

A corridor employing a fare system (FS) with VOD is considered, i.e., that recognizes users at the origin and destination stop. It is possible to build a database of

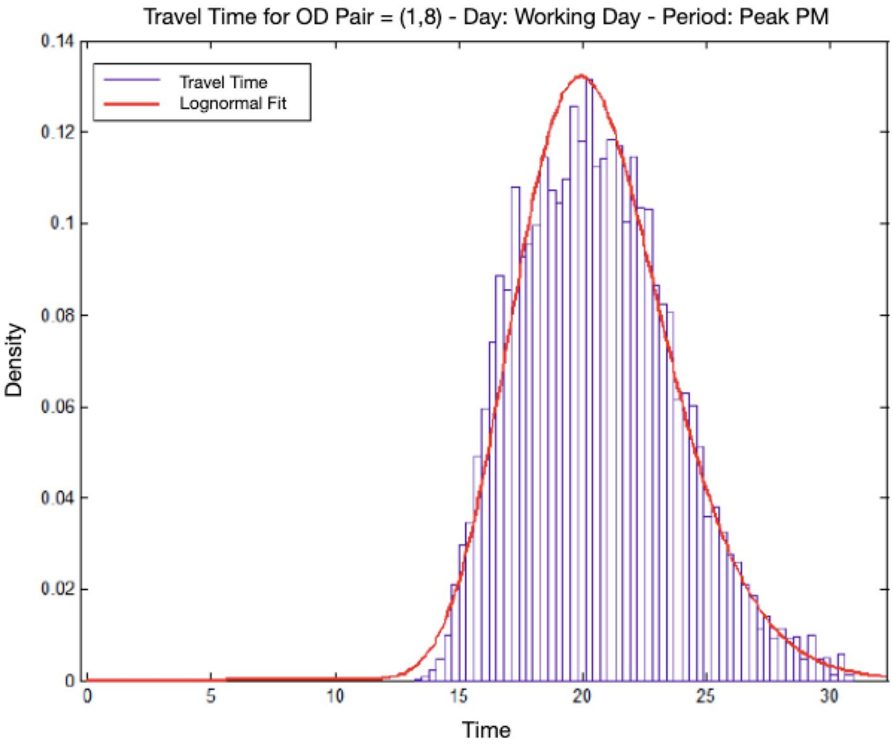

Fig. 1 OD pair (1,8) lognormal travel time distribution

trips made in the corridor, indicating the OD pair for each user, and their times of entry and exit, from whose difference experienced travel time is obtained.

In summary, operating with a FS with VOD allows for any temporal aggregation to have data such as:

Passenger entry counts for all stops.
Times of arrival of passengers at all stops.
Exit counts by OD pair.
OD matrices, obtainable from the database of trips.
Travel times for the OD pair between validations.

Analyzing historical travel time data from the base corridor, lognormal distributions are found to be the most suitable distributions for modeling travel time across the system.

Empirical travel times for every OD pair in the system were studied and the parameters for every lognormal distribution were calculated, and stored as historical data. For instance, empirical information and the fitted lognormal distribution for one specific OD is shown in Fig. 1.

The parameters that fit observed travel times distribution are not updated in this investigation. We recommend updating them every time the operator performs a major change in frequencies.

3 Methodology

Section 3.1 introduces the methodology to estimate the parameters of the travel time probability distribution in the corridor with incomplete information. Then, Sect. 3.2 presents the Bayesian modeling to estimate the travel destination originated at each stop. In Sect. 3.3 basic concepts regarding artificial neural networks used in this article are detailed.

3.1 Parameter estimation with incomplete information

In the public transport system considered, the system state is observed only every Δ minutes. At those moments, some trips are observed which are initiated but not ended, so it is not known where or when they will end. Therefore, one has incomplete information or censored data, which must be estimated. The destination stop of these trips will be known at some later moment than the current time of observation k^{+}, which is a multiple of Δ and corresponds to the end of interval k. In these cases, we say that the unobserved data are censored from the right.

A methodology for calculating maximum likelihood estimates (MLE) in scenarios where there is data censorship from the right (or of type I) follows, based on Patti et al. (2007), and adapted to the case of a corridor.

A first need is to estimate a distribution function for travel times between OD pair $w = (i,j)$ in the corridor. At any moment in time one could have incomplete information because travelers making a specific trip may have not left the system yet.

We will call $f(u^{w})$ to the probability density function of travel time between pair w, $F(u^{w})$ to its cumulative distribution function, and $S(u^{w}) = 1 - F(u^{w}) = P(U > u^{w})$ to the survival function. If we call θ^{w} to the vector of parameters associated with $f(u^{w})$ and $S(u^{w})$, the log-likelihood function l^{w} associated with the calibration of these parameters can be approximated as follows:

$$l^{w} = \sum_{r=1}^{n_i} \left[\delta_r \cdot \ln\left(f(\theta^{w})\right) + \left(1 - \delta_r\right) \cdot \ln\left(S(\theta^{w})\right)\right], \tag{1}$$

where n_i is the number of individuals who entered the system at station i, and $\delta_r = 1$ if the travel time of the r-th individual is known exactly, and $\delta_r = 0$ if such time is censored (not yet available). This is an approximation since we are considering that all passengers entering at a station i during a time interval will have the same travel time to any destination station j, which will naturally vary according to the OD pair w.

It is assumed that the data is under simple type I censoring. Thus, it can be assumed that up to time v the travel time of the first b users is known, and that the remaining $\left(n_i - b\right)$ are censored. Therefore:

$$\delta_r = \begin{cases} 1, & i \le b \\ 0, & i > b \end{cases}. \tag{2}$$

Incorporating these assumptions, rewriting expression (3):

$$l^w = \sum_{r=1}^{b} \ln\left(f(\theta^w)\right) + \left(n_i - b\right) \cdot \ln\left(S(\theta^w)\right). \tag{3}$$

Then, to find the MLE nonlinear expression (3) should be maximized, considering the sign constraints for parameters θ^w.

3.2 Model of travel distribution

3.2.1 Proposed model

We assume that the travel destination D_i^k of users starting their trips at stop i during interval k, is a random variable with multinomial distribution $D_i^k \sim MN\left(y_i^k;q_i^k\right)$, where y_i^k represents the number of trips originating at stop i during interval k, and $q_i^k = \left\{q_{ij}^k\right\}_{j=1}^N$ is the vector of probabilities of choosing j as travel destination. Therefore, the elements of vector D_i^k must add up to y_i^k.

According to Bayesian statistics (see, for example, Maritz and Lwin 1989), the conjugate *prior* distribution for the multinomial distribution is a Dirichlet distribution, thus vector q_i^k is considered to distribute Dirichlet with parameters $\alpha_i^k = \left\{\alpha_{ij}^k\right\}_{j=1}^N$, or $Dir\left(\alpha_i^k\right)$. As new information becomes available, vector q_i^k will be updated using Bayes' theorem. Then, being a conjugate distribution, the posterior distribution will be $Dir\left(\hat{\alpha}_i^k\right)$, in which $\hat{\alpha}_i^k \neq \alpha_i^k$. The new vector $\hat{\alpha}_i^k$ is calculated using the methodology proposed by Wicker et al. (2008).

3.3 Updated Bayesian parameter

The dynamic OD matrices of the corridor are formed by terms x_{ij}^{kh} of expression (4), which represent the estimate in interval h of the number of trips between each pair (i,j) initiated during interval k.

For estimating/updating the dynamic OD matrices of the corridor, matrices M_i^{kh} are used. They are formed by m_{ij}^{kth} elements, which represent the information updated up to interval h of the number of trips started at stop i during interval k, which ended (or will end) at stop j during interval t. Note that these elements are related to how users are distributed across the network and in time (i.e. how users get out of the system), and will be continuously updated using data of station and times in which users enter and exit the system, for all time intervals. The prior distribution for M_i^{kh}

is obtained using historical information for both travel destination and travel time distributions.

The terms x_{ij}^{kh} are calculated summing m_{ij}^{kth} elements, which contain an observed part ($t \in [k, h]$) and an estimated part ($t > h$), which together allow to generate every new estimate/update of the dynamic OD matrices of the corridor.

The m_{ij}^{kth} elements are obtained in a 2-step procedure: (1) estimating the number of trips originating at station i that will be coming out of the system in intervals $t > h$, and (2) distributing these trips to all different destinations in each interval t.

The outcome of the first step is obtained by multiplying the number of trips originated at station i during interval k (y_i^k) and the proportion of those trips that should be coming out of the system in intervals $t > h$. This proportion is updated using information available up to interval h, of where users exit the system.

The second step distributes the number of trips obtained in the first step across all destinations and in the remaining time intervals ($t > h$), using the updated probability at interval h of a trip starting at stop i in interval k that ends at stop j in interval t.

Then, the updated probability at interval h (posterior mean) that a trip that started at stop i during interval k had stop j as destination, is given by the following expression:

$$q_{ij}^{kh} = \frac{\alpha_{ij}^k + x_{ij}^{kh}}{\alpha_{i0}^k + y_i^k}, \forall i, j. \tag{4}$$

In expression (4) the term $\alpha_{i0}^k = \sum_j \alpha_{ij}^k$. Note that the above expression is used to leverage the knowledge of the historical distributions of travel destinations in interval k, stored in parameters α_{ij}^k. Thus, in situations where, for example, the output process of passengers from stops is delayed beyond normal, one will avoid assuming immediately—and wrongly—that such delay is due to a change in the structure of the distribution of travel destinations. To infer whether that change actually occurred, we must resort to more information, such as travel times.

3.4 Artificial neural networks

Artificial neural networks (ANNs) use a basic processing unit, called artificial neuron (AN). Within ANNs, knowledge is acquired after a learning process, and stored as "synaptic weights", which correspond to the connections of the ANN.

An ANN is a group of artificial neurons interconnected by several links. The architecture of these networks corresponds to the ordering that is made from such set of neurons. Grouping neurons is typically performed in layers, in which the input data is processed—input layers—internal calculations are performed—hidden layers—and the outputs of the ANN are generated—output layer. Furthermore, each layer contains a certain number of artificial neurons.

The number of neurons n_{in} in the input layer corresponds to the number of input parameters of the ANN, and is given by the size of the input data vectors. Likewise, the number of neurons n_{out} in the output layer is given by the size of the output vectors.

Meanwhile, both the number of hidden layers and the number of neurons n_h in each one of them must be calibrated heuristically, since there is not an optimum value, and it depends on the problem being treated. Cybenko (1989) showed that the ANNs possess the property of "universal estimators", i.e., an ANN can approximate any real function to an arbitrary level of precision. In order to achieve this, it is required that the ANN has an architecture that includes a hidden layer, with "sufficient" neurons therein. Therefore, architectures with just one hidden layer were considered, and the number of neurons n_h in it was to be determined.

The topology and the shape that the ANN finally has is given by the connections that are made within and among the layers. When there are links that are only forward—i.e., from the input layer to the output layer—one speaks of a feedforward network. Meanwhile, when there are links between neurons in the same layer or previous layers, it is called recurrent networks.

3.4.1 Supervised neural networks

Supervised neural networks are one type of ANN where one seeks that the outputs it generates are as similar as possible to the actual values available, which is achieved by adjusting its parameters, in a process known as ANN training.

The aim of the training is to minimize an error function that relates the outputs delivered by the network and the actual values. Prior to the start of training, preprocessing of the data must be performed, normalizing it to the range corresponding to the co-domain of the activation functions used. Then, the normalized database is divided into three sets:

1 Training data (TrD): Progressive adjustment of network parameters.
2 Validation data (VD): Evaluation of the generalization ability of the network.
3 Test data (TeD): Independent measurement of network performance.

Training consists of two phases, which are detailed in Zúñiga (2011):

- Feedforward phase: aims to present and propagate through the ANN the couples of inputs and outputs.
- Backpropagation phase: aims to estimate the changes the synaptic weights will experience during training.

3.4.2 Selection of the ANN architecture

The most suitable architecture for a particular problem is related to the adjustment level you get while using such structure in training an ANN. There is, however, a wide range of possible architectures for the ANN, which are obtained from the different values that the vector $\left(n_{in}, n_h, n_{out}\right)$ can take. Then, for a given architecture, the

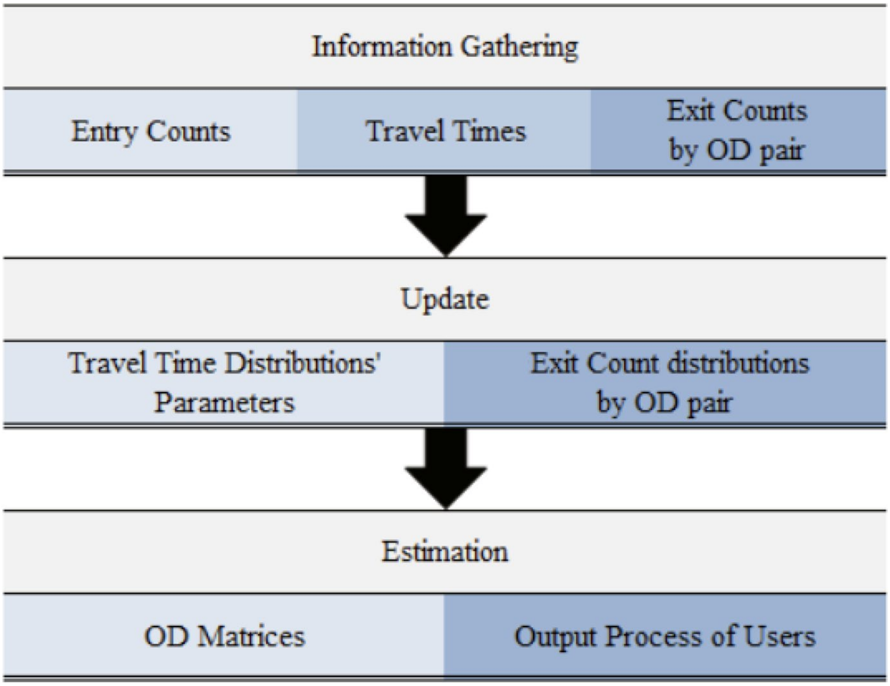

Fig. 2 General outline of the estimation algorithm

output delivered by the ANN—after training the ANN a low number of epochs (i.e. one cycle through the full training dataset)—is compared to TeD data set, and the goodness of fit is calculated. Heuristically, a low number of epochs is used to accelerate the process of selecting the architecture.

The adjustment measure to be used is the weighted information criterion (WIC), defined and used by Eğrioğlu et al. (2008). The network architecture that minimizes the WIC is the one that best fits the TeD set, and at the same time keeps its complexity controlled, to maintain excellent generalization ability and avoid the phenomenon of overfitting.

4 Estimation of OD matrices for public transport (EODPT)

4.1 General description of the EODPT algorithm

The EODPT algorithm performs successive estimates of the OD matrices of the corridor under analysis, through the progressive incorporation of the information available up to h^+ (the end of interval h), particularly exits per stop and real travel times before delivering the final estimate associated with some interval of the study period.

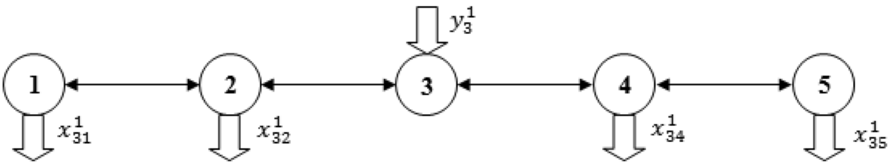

Fig. 3 Scheme of test corridor—algorithm of EODPT

Table 1 Scenario matrix

Case	$q_{ij}^{kh} = q_{ij}^{k0}$	$f_{ij}^{kh}(\cdot) = f_{ij}^{k0}(\cdot)$
1	✓	✓
2	✓	×
3	×	✓
4	×	×

Figure 2 shows the overall scheme of operation of the proposed algorithm. In a first step, information is collected from VOD. This data feeds the second general step of the algorithm, where different internal parameters of the current model are updated, leading to the estimation of OD matrices of intervals associated with new information. This update includes the OD matrices for every trip of the corridor to interval h, inclusive, since many of them have still not ended.

Note that the incorporation of observed travel times allows updating all travel time distributions. With this new information it is possible to update the probabilities related to how users travel across the network and in time. The latter probabilities p_{ij}^{kth} are contained in matrices $P_i^{kh}[j, t]$, and they represent the probabilities estimated in interval h that a trip originated at stop i during interval k ends at stop j during interval t.

The EODPT algorithm consists mainly of three steps. First, travel time distributions $f_{ij}^{kh}(\cdot)$ are updated, solving the MLE problem (3) applied to this case. Secondly, using these new travel time distributions, the probabilities p_{ij}^{kth} are updated according to the actual time of arrival of the passengers who entered the system at stop i during interval k. Thirdly, the probabilities p_{ij}^{kth} and the information of how users exit the system in each time interval allows us to calculate the q_{ij}^{kh} terms of Eq. (4), which end up in the new estimation of the OD matrices of the corridor.

4.2 Analysis of estimation results

The performance of the EODPT algorithm was evaluated considering a hypothetical corridor under different levels of abnormality between the historical

Table 2 Travel distribution parameters q_{3j}^{k0} for each of the four cases: proportion of trips from node 3 to each node j

Case	Destination stop j			
	1	2	4	5
1 and 2	0.3	0.1	0.2	0.4
3 and 4	0.1	0.3	0.4	0.2

distribution of travel destinations and the observed situation. The test also included a congestion effect to cause the trips to lengthen, and therefore, the output processes of users should take longer than normal.

4.2.1 Travel estimation for test corridor

A small fictitious corridor composed of only five stops as shown in Fig. 3 was considered. All trips begin at the third stop and head to the other four.

For this corridor four cases were considered, which involved different levels of abnormality between the historical situation and the observed one, both in the distribution of travel destinations and travel times. Table 1 schematically presents the characteristics of each case.

Cases 1 and 2 described in the matrix correspond to those where travel destinations show a very similar distribution to the historical one. The difference is that in the latter a congestion effect is observed, lengthening travel times with respect to the historical situation. In cases 3 and 4 the distribution of travel destinations is different from the historical one. The difference between cases 3 and 4 is again the congestion effect.

In all 4 cases, the times of arrival of passengers within each interval are identical. These were generated randomly using a uniform distribution.

4.2.2 Parameters used

In the test corridor, it was considered that during interval $k = 1$, to stop $i = 3$ enter a total of $y_3^1 = 200$ trips. In addition, it was assumed that the maximum duration of a trip on the network, S_{max}, corresponds to 6 intervals.

First of all, a prior probability distribution of exits p_{ij}^{kt0} needs to be determined (associated with the interval 0) to predict how many of those 200 trips will end at stop j during interval t. For this, the specific time of arrival of each user is used and a lognormal distribution of travel time for trips between i and j initiated during interval k is assumed, defined by historical parameters θ_{ij}^{k0}.

In Table 2 the parameters of the trip distribution model are summarized, used to generate each test case. It is important to remember that the parameters associated with cases 1 and 2 are assumed to be available as historical information for cases 3 and 4.

For each of the four test cases, 100 different instances of 200 trips were generated and distributed to the four destinations according to their associated probabilities.

Table 3 Average distribution of exits by interval and stop for Case 4

j	Sum	*h*					
		1	2	3	4	5	6
1	20.11	0.00	1.98	10.06	8.07	0.00	0.00
2	60.76	19.74	41.00	0.02	0.00	0.00	0.00
4	80.05	38.69	41.36	0.00	0.00	0.00	0.00
5	39.08	2.27	30.78	6.03	0.00	0.00	0.00
% Information		30.35%	87.91%	95.97%	100%	100%	100%

Table 4 Estimation error using the historical average trip distribution: average and standard deviation of the error for each performance indicator

Historical		
SRMSE	RMSE	RNWP
2.0410 (0.0351)	16.3279 (2.2464)	0.8081 (0.0058)

The arrival times of the two hundred people are the same in each of the 100 instances and also identical for the four analyzed cases. Finally, travel times are randomly generated from a lognormal distribution, different for each OD pair, using the parameters reported in Zúñiga (2011).

4.2.3 Results

For brevity, we focus in this section only on case 4 which is the most difficult to forecast, knowing that for the other three cases the performance of the algorithm is better than for this case. Table 3 shows the average distribution (over 100 instances) of the exit interval and stop of the 200 trips.

We utilized three error measures to assess the performance of any predictive method: root mean square error (RMSE), the standardized root mean square error (SRMSE) and the relative number of wrong predictions (RNWP). We compute the average performance indicator and the standard deviation for each one.

We use the historical average trip distribution as our dynamic OD matrix estimation benchmark to our method. Table 4 reports its performance.

The performance of the EODPT algorithm is presented in Table 5. It shows the evolution of the average error obtained at the end of each interval or iteration through the EODPT algorithm, and the average calculation times and confidence intervals (CI) at 95%.

It should be noted that with just 30% of the data at the end of the first interval, a 5% improvement is achieved over the estimate using historical information. Then, after interval $h = 2$, where there is already almost 88% of information, close to 92% improvement is obtained with respect to the historical information, i.e., the benefits are rapidly seen. The evolution of the estimated probabilities of trip destination choice is shown in Fig. 4. It is observed that the probabilities are updated rapidly after the first iteration without reaching the real observed value in this case, as the historical probability distribution remains with some weight.

Table 5 Error evolution using the EODPT algorithm: average and standard deviation for each performance indicator after each interval h

h	Error measure			Improvement percentage			Computation time (s)	
	SRMSE	RMSE	RNWP	SRMSE (%)	RMSE (%)	RNWP (%)	Average	CI
1	1.9374 (0.0202)	15.4993 (1.2906)	0.7684 (0.0035)	5.07	5.07	4.91	1.0402	[1.0056; 1.0747]
2	0.1744 (0.0087)	1.3950 (0.5600)	0.0494 (0.0007)	91.46	91.46	93.89	5.2271	[4.5517; 5.9024]
3	0.0738 (0.0013)	0.5903 (0.0830)	0.0209 (0.0001)	96.38	96.38	97.42	2.062	[1.9848; 2.1392]
4	0.0000 (0.0000)	0.0000 (0.0000)	0.0000 (0.0000)	100	100	100	1.8856	[1.7751; 1.9961]

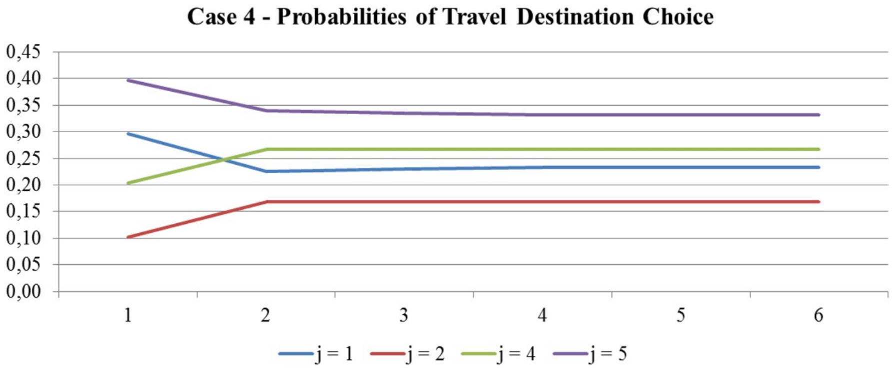

Fig. 4 Probabilities of travel destination choice

Table 6 Variables used in the PODPT algorithm

v_1	Day of the week
v_2	Interval k of the day
v_3	Number of trips between (i, j) pair in the same interval k in the last R days of the same type
v_4	Number of trips between (i, j) pair in the last p intervals of the current day. In this case, a combination of actual data and estimates of EODPT algorithm will be available
v_5	Calendar information, such as if the current day is a holiday or not
v_6	Presence of any particular event, which are extracted from historical data. Within the database, the following events were found: 1. Soccer match 2. External strike on Merval: A strike in the services of buses in Valparaiso was observed, which generated an increase in the number of trips in the corridor 3. Internal strike on Merval: Merval's train operators strike produced a lower supply in the corridor, which generated a decrease in the number of trips in the corridor 4. Cruise arrival: Cruise Queen Mary 2 arrived in Valparaiso's port, generating a change in the pattern of trips in the corridor

To improve the operations of a public transport system, we need not just to have a good understanding of the OD matrices that explain the flows that the system is observing, but also to improve the prediction of future trip patterns. In the next section we use the PODPT algorithm to predict future flows for a real Metro line.

5 Application of the prediction method for OD matrices for a corridor

A test was performed on the proposed method for estimating and forecasting OD matrices, using the information available for the corridor of the Metro of Valparaiso, Merval.

This chapter presents an application of the proposed algorithm for the PODPT with VOD using ANNs. The objective is to predict at the end of interval h, the number of trips between each OD pair in the corridor, to begin at later intervals, i.e. $k > h$. The following sections describe the methodology that defines the input data, the architecture that the ANN will have, and the results obtained.

5.1 Description of the corridor and variable selection for the PODPT algorithm

The data used in this process is related to the operation of the corridor of Metro of Valparaiso (Merval) during the first 6 months of the years 2006, 2007 and 2008. This corridor serves 20 stops, crossing from the port city of Valparaiso to Limache, through the urban area of Viña del Mar, Chile. The information was previously grouped into three types of day: workday, Saturday and Sunday or holiday, and analyzed independently. Therefore, the kind of day will not be part of the set of variables that will influence x_{ij}^{k}—the number of trips between the pair (i, j), initiated during interval k. For a particular type of day, the variables that were considered to influence x_{ij}^{k} (i.e. the inputs) are summarized in Table 6.

The ANN method predicts (i.e. the output), at the end of the interval h, the next s values of the time series of x_{ij}^{τ}, where $\tau = h + 1, \dots, h + s$. The proposed methodology would allow the recursive use of the predictions obtained at each interval of the day to feed back the model and generate a forecast of the variables x_{ij}^{τ} with $\tau > h + s$, that is, for all remaining intervals of the day. However, in this application the model is applied to predict interval t after $h + s$ only once the actual demand up to interval $t - s$ is known.

5.2 Preprocessing and coding data

The ANN feedforward type used the hyperbolic tangent as activation function in each of the artificial neurons of the input hidden layers, while in the output layer the identity function was used. As part of the preprocessing, all data were normalized at the interval $[-0, 9;0, 9]$ to avoid saturation of the first function, i.e., to avoid the more extreme values of the curve. During training, performed with the *Neural Network Toolbox* of Matlab™, standard division of the data set was performed, distinguishing the TrD set, the VD set and the TeD set.

5.3 Procedure to define the architecture of the network

As already mentioned, the architecture of the network is related to the parameter vector (p, R, s). The parameters p and R, together with the coding used (i.e. the variables used), generate a total of $n_{in} = (p + R + 4)$ input parameters. As for s, it represents the number of output parameters n_{out} of the network.

As for the parameter n_H, according to an analysis of De La Fuente García (1995), it is recommended that this value should be at least 75% of n_{in} (Salchenberger et al. 1992), and can reach up to $\left(2 \cdot n_{in} + 1\right)$ (Zaremba 1990). An ANN was used with the following architecture:

1 An input layer with n_{in} neurons.
2 A hidden layer with n_H neurons.
3 An output layer with s neurons.

The value of the variables in vector (p, R, s) was varied, obtaining different values for n_{in}, for which the number of neurons n_H was varied within the suggested range. The number n_H^* of neurons—associated with the vector (p^*, R^*, s^*)—will be the one that minimizes the WIC, during a training conducted for 20 epochs. Finally, the final structure will contain $n_{in}^* = (p^* + R^* + 4)$ neurons in the input layer, n_H^* neurons in the hidden layer and s^* neurons in the output layer, and a full training will be performed using this architecture.

5.4 Results

The experiment performed and the results obtained in the selection of the architecture of the ANN are described in the following section. Subsequently, the results of the prediction with the final structure are shown. In both cases, the data of the trips made in 2007 and 2008 were used, which were divided into TrD = 85%, VD = 10% and TeD = 5%.

5.4.1 Selection of the ANN architecture

For this experiment, the data of only three OD pairs was used—(1,8), (8,1), (20,8)—which correspond to pairs of stops which were considerably affected by the occurrence of events highlighted in Table 6. The range of variation of the vector (p, R, s) is given by the following values: predictions were considered only one step ahead, i.e., $s = 1$, p between 1 and p_{max} dependent of the OD pair, and R between 1 and 4.

The p_{max} value comes from an analysis of the partial autocorrelation function (PAF) of the data. From PAF we extract the first lags of the series with the greater statistically significant participation in the explanation of the current value, if a linear autoregressive model was used to explain the series. The range for R emerges to include similar periods of several weeks before the prediction.

Table 7 Results of ANN architecture selection

Type of day	Origin	Destination	WIC				RMSE			
			Minimum	(p, R, s)	n_h	Parameters	Minimum	(p, R, s)	n_h	Parameters
Workday	1	8	0.1832	(9, 1, 1)	10	150	0.5289	(13, 2, 1)	29	580
	8	1	0.2913	(13, 3, 1)	21	441	0.4349	(17, 3, 1)	21	525
	20	8	0.1572	(15, 4, 1)	17	408	0.7422	(15, 4, 1)	29	696
Saturday	1	8	0.168	(8, 4, 1)	12	192	0.4248	(16, 3, 1)	17	391
	8	1	0.2326	(8, 3, 1)	14	210	0.4245	(19, 3, 1)	28	728
	20	8	0.2961	(10, 2, 1)	12	192	0.7207	(15, 2, 1)	21	441
Sunday or holiday	1	8	0.2395	(12, 2, 1)	13	234	0.3603	(12, 4, 1)	18	360
	8	1	0.2127	(10, 4, 1)	19	342	0.3535	(18, 2, 1)	24	576
	20	8	0.2575	(10, 4, 1)	13	234	0.7582	(15, 1, 1)	18	360

Table 8 Error measure of the PODPT algorithm: average and confidence interval for each type of day

Type of day	Origin	Destination	RMSE	
			Average	CI
Workday	1	8	0.5598	[0.5525; 0.5670]
	8	1	0.4419	[0.4392; 0.4446]
	20	8	0.7246	[0.7177; 0.7314]
Saturday	1	8	0.4252	[0.4188; 0.4315]
	8	1	0.431	[0.4236; 0.4384]
	20	8	0.807	[0.7992; 0.8148]
Sunday or holiday	1	8	0.4208	[0.4125; 0.4291]
	8	1	0.3635	[0.3558; 0.3712]
	20	8	0.784	[0.7754; 0.7927]

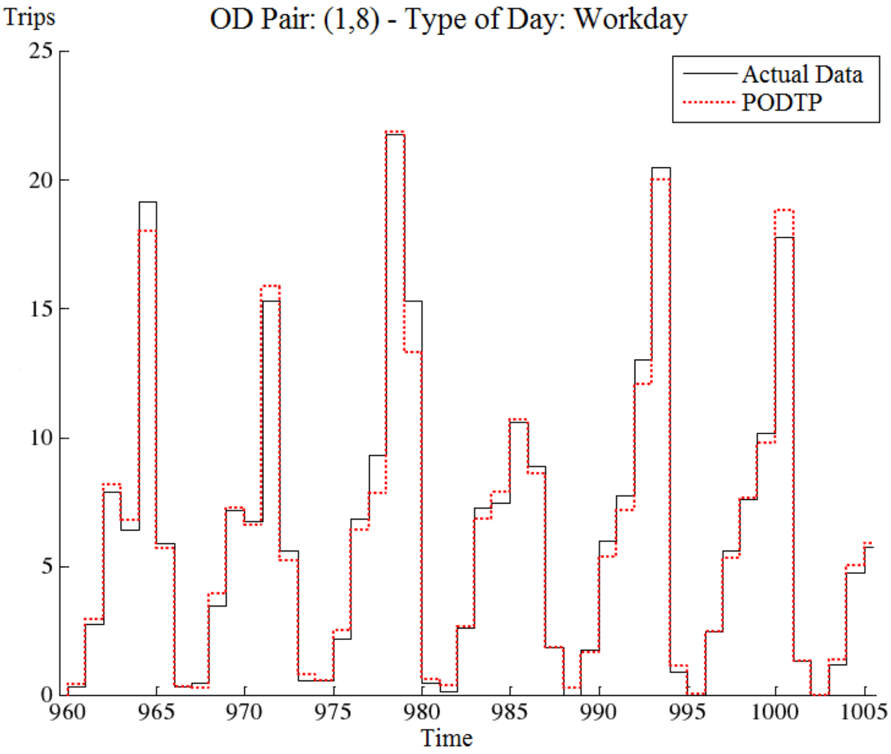

Fig. 5 PODPT predictions—pair (1,8)—weekday

The training was conducted for only 20 epochs, to accelerate the process of selecting the most appropriate architecture. It is important to remember that the number of possible architectures could be very large, because it is derived from all possible values of the vector (n_{in}, n_h, n_{out}). After the adequate architecture is selected a proper training is conducted, from which the actual forecasting model is derived. Then, for each value of s, the value of n_s that minimizes the WIC in the TeD group was sought. Then, the combination of (p, R, s) that generated this value n_s and the parameters of the chosen architecture were identified. The results were compared with those that would have been obtained using the RMSE as a selection criterion. The results of this process are shown in Table 7, distinguished by the type of day.

Table 7 shows that the architecture selected using the WIC criterion is always less complex than the one selected using the RMSE, since a structure with lower n_h and fewer total parameters is always chosen.

5.4.2 Training with the selected architecture

Training was conducted with the architectures selected in the previous section, with the same TeD set and the same methodology, but using a greater number of epochs (i.e. 1000). 30 different random initializations for the synaptic weights were performed and the results shown in Table 8 correspond to the average of those 30 cases—and the 95% confidence intervals—using the RMSE as error measure, comparing the predictions against the 5% of the observed dataset that was reserved for this purpose (i.e. the TeD).

A low average forecasting error in the TeD set was obtained. Furthermore, we see that the confidence intervals are quite narrow, which allows sustaining that the architecture used is appropriate, and that the predictions are reliable.

As example, Fig. 5 shows graphically the comparison (of some data) between the predictions of the ANN (in red) and the actual values (in black) for the pair (1,8) on weekdays, which show that the results are of high quality. It is also remarkable how the predictions manage to capture direction changes of the data, and how in addition, they remain very close to the actual values.

OD pair (1,8) was selected as one the OD pairs affected by the events that disturbed the normal demand behaviour; therefore, it was considered interesting to be analyzed.

6 Conclusions

It should be mentioned, first, that the proposed methodology explicitly incorporates a richer and broader set of information, associated with a corridor that uses a FS with VOD. Then, in the cases tested, both the PODPT and the EODPT algorithms are capable of delivering real-time results.

Both the speed and the quality of the results allow sustaining that this methodology can be used in a practical way, focusing its application to generate better estimates of future states of the system, in order to study the implementation of control measures, which may be incorporated into real time operation of the corridor.

Regarding the EODPT algorithm, the incremental adjustment of the results corresponds to a Bayesian approach in its modeling, which typically is not considered in OD matrices update processes. It is shown that the use of this approach and the incorporation of new information improve the results when compared to the historical method of updating, both in similar and dissimilar scenarios with respect to the historical situation. It was observed that it performs very well in different scenarios, which differed from each other in the level of abnormality regarding the historical trips distribution. In the test cases, computation times remained low, not exceeding 10 s.

The speed of the EODPT algorithm depends on the amount of available information. If this is not enough, the historical estimate remains the best available. Otherwise, the algorithm searches for a solution to the proposed optimization problem.

Regarding the PODPT algorithm, the calibration of the ANNs must be performed offline. Once ANNs are trained, the results are delivered in real time, also with a good generalization capacity in the face of new scenarios. Section 5.4.2 shows how high quality results are obtained in the predictions, due to the methodology used to select the most appropriate architecture. Of this selection scheme the compromise between accuracy and complexity must be highlighted, which ensures that selected ANNs have a good generalization capacity, along with a good ability to fit the data used during training.

It was also observed that the used set of variables, corresponding to all the information that was available, was sufficient to find ANN architectures that performed well in the predictions. Additionally, it may be mentioned that estimates from the EODPT algorithm can be used for variables that are related to trips between each OD pair in the corridor. As a future work, we should be exploring recurrent neural networks and reinforcement learning techniques.

Finally, an important discussion relates to computation times. If these values should be kept low, it is important to review the network size, perform offline ANN calibrations and revise the hardware configuration. Note also that MLE problems may not always converge to a solution, which is approximated by an internal iterative process.

The methodology presented in this paper could be used for real-time control actions. It could become a key input for advanced operational policies aiming at headway regularity as train injection, speed control or holding at stations. These strategies yield more even headways which reduce unreliability in waiting and crowding conditions. The strategy can also be quite timely for a sanitary crisis as COVID19, in which public transport operators are requested to guarantee their users a minimum distance between passengers on the platforms and inside the trains. This would require metering the number of passengers allowed to board each train so that crowding conditions inside each one do not exceed the desired sanitary recommendations. An accurate real-time prediction of the origin-destination matrix of the arriving passengers would be desirable to implement the needed metering strategy in each station.

Acknowledgements This research was supported by the Bus Rapid Transit Centre of Excellence funded by the Volvo Research and Educational Foundations (VREF). The authors also gratefully acknowledge the research support provided by CEDEUS, ANID/FONDAP 15110020 and FONDECYT 1171049.

References

Ashok K, Ben-Akiva M (1993) Dynamic origin–destination matrix estimation and prediction for real-time traffic management systems. In: Proceedings of the 12th international symposium on traffic and transportation theory (1993). Berkeley July 1993, pp 465–484

Bierlaire M, Crittin F (2004) An efficient algorithm for real-time estimation and prediction of dynamic OD tables. Oper Res 52(1):116–127

Carrese S, Cipriani E, Mannini L, Nigro M (2017) Dynamic demand estimation and prediction for traffic urban networks adopting new data sources. Transp Res Part C Emerg Technol 81:83–98

Cybenko G (1989) Approximation by superpositions of a sigmoidal function. Math Control Signal Syst 2:303–314

De La Fuente García D, Pino Díez R (1995) Comparison of estimates and calculations of univariate transfer function, using the methodologies Box–Jenkins and neural networks. Qüestiió 19(1):187–215

Delgado F, Muñoz JC, Giesen R, Cipriano A (2009) Real-time control of buses in a transit corridor based on vehicle holding and boarding limits. Transp Res Rec 2090:59–67

Delgado F, Muñoz JC, Giesen R (2012) How much can holding and/or limiting boarding improve transit performance? Transp Res Part B Methodol 46(9):1202–1217

Eğrioğlu E, Hakan Aladağ Ç, Günay S (2008) A new model selection strategy in artificial neural networks. Appl Math Comput 195:591–597

Jenelius E (2019) Data-driven metro train crowding prediction based on real-time load data. IEEE Trans Intell Transp Syst 21(6):2254–2265

Krishnakumari P, van Lint H, Djukic T, Cats O (2020) A data driven method for OD matrix estimation. Transp Res Part C Emerg Technol 113:38–56

Li B (2008) Markov models for Bayesian analysis about transit route origin–destination matrices. Transp Res Part B Methodol 43:301–310

Maritz JS, Lwin T (1989) Empirical Bayes methods. Chapman & Hall, London

Nguyen S, Pallottino S (1986) Estimating origin destination flows for transit networks. Ricerca Oper 38:9–27

Nguyen S, Morello E, Pallottino S (1988) Discrete time dynamic estimation model for passenger origin/destination matrices on transit networks. Transp Res Part B Methodol 22:251–260

Patti S, Biganzoli E, Boracchi P (2007) Review of the maximum likelihood functions for right censored data. A New Elementary Derivation. COBRA Preprint Series, Article, p 21

Rahman S, Wong J, Brakewood C (2016) Use of mobile ticketing data to estimate an origin-destination matrix for New York City Ferry Service. Transp Res Rec 2544(1):1–9

Salchenberger LM, Cinar EM, Lash NA (1992) Neural networks: a new tool for predicting thrift failures. Decis Sci 23:899–916

Sherali HD, Park T (2001) Estimation of dynamic origin–destination trip tables for a general network. Transp Res Part B Methodol 35:217–235

Toqué F, Côme E, El Mahrsi MK, Oukhellou L (2016) Forecasting dynamic public transport origin-destination matrices with long-short term memory recurrent neural networks. In: 2016 IEEE 19th international conference on intelligent transportation systems (ITSC) (pp 1071–1076). IEEE

Wicker N, Muller J, Kalathur R, Kiran R, Poch O (2008) A maximum likelihood approximation method for Dirichlet's parameter estimation. Comput Stat Data Anal 52:1315–1322

Wong SC, Tong CO (1998) Estimation of time–dependent origin–destination matrices for transit networks. Transp Res Part B 32:35–48

Zaremba T (1990) Technology in search of a buck. Neural network PC tools. Academic Press Inc, New York

Zhang J, Shen D, Tu L, Zhang F, Xu C, Wang Y, Li Z (2017) A real-time passenger flow estimation and prediction method for urban bus transit systems. IEEE Trans Intell Transp Syst 18(11):3168–3178

Zhou X, Mahmassani H (2007) A structural state space model for real-time traffic origin–destination demand estimation and prediction in a day-to-day learning framework. Transp Res Part B Methodol 41:823–840

Zúñiga F (2011) Estimation and prediction of dynamic matrix travel on a public transport corridor, using historical data and information in real time. Thesis of Master of Science in engineering, Department of Transportation Engineering and Logistics, Pontificia Universidad Católica de Chile